Dicentral

DiUnite 101

Copyright and Legal Disclaimers

User Manual for DiUnite

Copyright © 2011 by DiCentral Corporation

All rights reserved. This document and all content that is contained, embedded and linked to this document, and that is written by DiCentral Corporation, or its partners or affiliates is copyright and bound by all copyright laws.

This document is the property of DiCentral Corporation and/or its affiliates and contains public information about proprietary products and/or services. All information contained herein is the property and exclusive copyright of DiCentral Corporation. No information within this document, nor information about the products or services described in this document, may be distributed or copied without the expressed written consent of DiCentral Corporation.

For further information, contact DiCentral Corporation.

Address:	1199 NASA Parkway, Houston, TX 77058
Phone:	+1.281.480.1121 or +1.713.559.0580
Fax:	+1.281.480.1181
Website:	www.dicentral.com

Table of Contents

Copyright and Legal Disclaimers **0**

Chapter I: Introduction **5**

1. Introduction 5

2. System Requirements 6

Chapter II. Creating a Document **7**

1. Documents 7

2. Starting Document Manager 7

3. Creating Source and Target Documents 8

4. Creating a Map 18

4.1 Starting Data Link Manager 18

4.2 Mapping from a Source Document to an Exchange Document 19

4.3 Mapping from an Exchange Document to a Target Document 33

5. Creating a Job 38

5.1 Starting Job Manager 38

5.2 Creating a New Job 39

5.3 Setting up the Job 40

Appendix A: The Functions **45**

1. Data Manipulation 45

1.1 Abs [] 45

1.2 AddDate [name | value | datepart] 45

1.3 Convert [datatype | format] 45

1.4 DefaultValue [value] 45

1.5 DelimitedMid [value | start | end | fieldseparator | fieldno | key | keyseparator | outputvariable] 45

1.6 Find [string | position | startposition | replace | replacestring] 46

1.7 FixedValue [value] 46

1.8 Format [format] 46

1.9 InsertDelimiter [sections | value | start | end | fieldseparator | key | keyseparator] 46

1.10 Left [length | value] 47

1.11 Length [value] 47

1.12 LowerCase[variable] 47

1.13 Mid [position | length | value] 47

1.14 Now [] 47

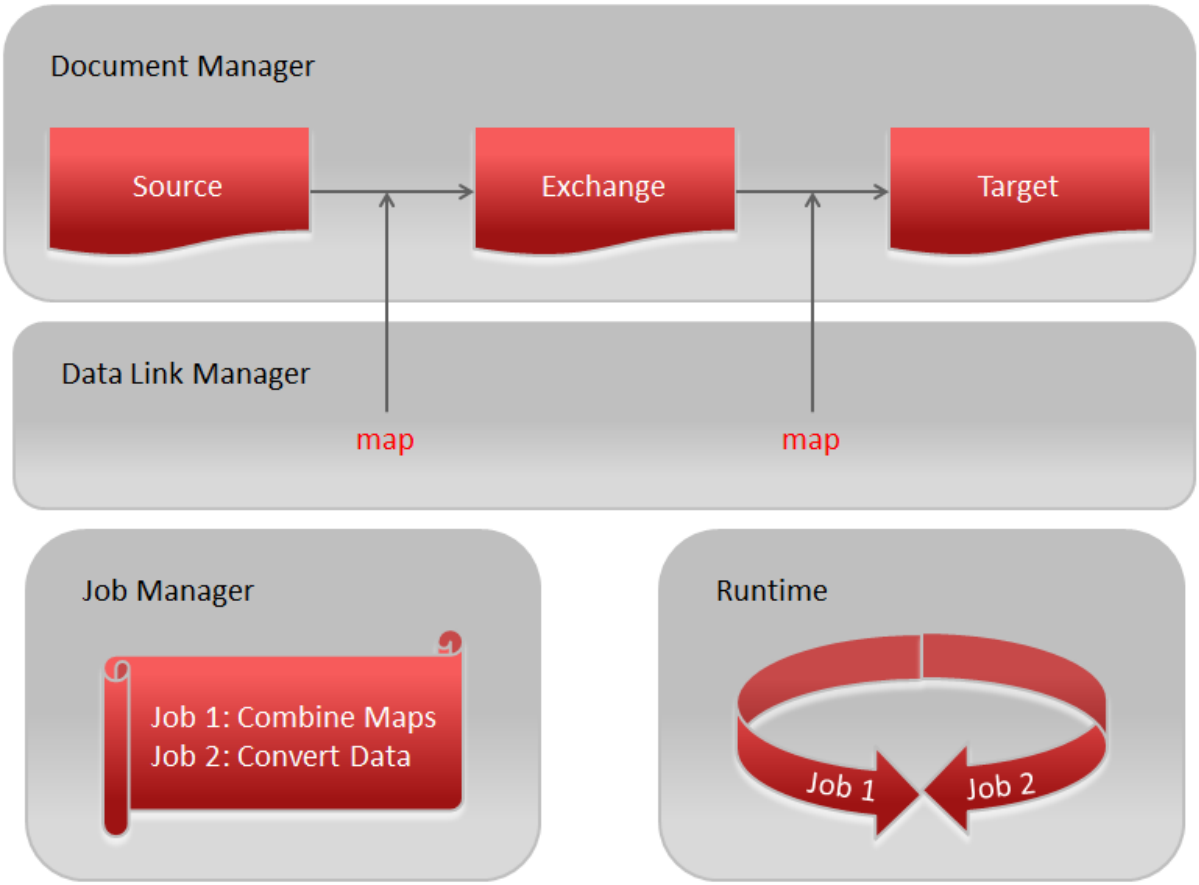
1.15 Padding [length | side | symbol | variable] 47
1.16 Right [length | value] 47
1.17 Trim [value | side] 48
1.18 UpperCase[variable] 48
2. Source Mapping 48
2.1 NewRow [exd | table] 48
2.2 SetValue [exd | table | field | value | template] 48
2.3 RemoveDuplicateRow [exd | table | field1 | field2 | field3 | field4 | field5 | sumField1 | sumField2 | sumField3 | sumField4 | sumField5] 48
2.4 RemoveEmptyRow [exd | table | field1 | field2 | field3 | field4 | field5] 49
3. Target Mapping 49
3.1 NextRow [exd | table] 49
3.2 GetRow [exd | table | field | value] 49
3.3 FindNextRow [exd | table | field | value] 50
3.4 GetValue [exd | table | field | aggregate] 50
3.5 ContinueMessage [] 50
4. Conversion Functions 50
4.1 AddConversion [fromvalue | tovalue | position] 50
4.2 AddStringConversion [list | fromvalue | tovalue] 51
4.3 StringConvert [list | default | casesensitive] 51
4.4 Clearconversion [] 51
4.5 Lookup [filename | section | casesensitive | load | key | default] 51
4.6 LookupAdv [filename | recordseparator | fieldseparator | keyfieldno | valuefieldno | casesensitive | load | key | default] 51
5. Exchange 52
5.1 SetFilter [exd | table | field | value | comparetype] 52
5.2 ClearFilter [] 52
5.3 ClearFilterAll [] 52
5.4 GroupTable [exd | table | field1 | field2 | field3 | field4 | field5 | sumField1 | sumField2 | sumField3 | sumField4 | sumField5] 52
5.5 ResetTable [exd | table] 53
6. Flow Control 53
6.1 AndIf [lvalue | rvalue | operator | datatype] 53
6.2 Break [] 53
6.3 ContinueLoop [] 53

6.4 Do [] 53
6.5 Else [] 53
6.6 ElseIf [lvalue | rvalue | operator | datatype] 54
6.7 EndIf [] 54
6.8 EndWhile [] 54
6.9 If [lvalue | rvalue | operator | datatype] 54
6.10 IfInList [filename | section| casesensitive | load] 55
6.11 Iif [lvalue | rvalue | operator | datatype | truepart | falsepart] 55
6.12 LoopWhile [lvalue | rvalue | operator | datatype] 55
6.13 OrIf [lvalue | rvalue | operator | datatype] 56
6.14 While[lvalue | rvalue | operator | datatype] 56
7. Variables 57
7.1 SetVariable [name | value | datatype] 57
7.2 OperateVariable [name | value | datatype] 57
7.3 GetVariable [name] 57
7.4 SetDynamicName [variable | format] 57
8. External resources 57
8.1 CreateGUID [] 57
8.2 GetUserLogin [] 58
8.3 GetCounter [name] 58
8.4 GetJobNo [] 58
8.5 GetPartnerValue [key] 58
8.6 SQLConnect [connectionname | connectionstring | userid | password | autocommit] 58
8.7 SQLDisconnect [connectionname] 58
8.8 SQLExec [connectionname | sql | fieldseparator | recordseparator] 59
9. Debug 59
9.1 Debug [variable | tracefile | traceall] 59
9.2 SaveExd [exd | filename] 59
9.3 LoadExd [exd | filename | category] 59
9.4 SetError [message | severity] 59
9.5 Validate [operator | rvalue | datatype | severity | message] 60
9.6 ValidateBetween [fromvalue | tovalue | datatype | severity | message] 60
9.7 ValidateRegex [regex | severity | message] 60

Chapter I: Introduction

1. Introduction

This tutorial contains step-by-step instructions for creating a document, creating a map, and creating a job. The result of the following procedures will be a job that, when run, will convert documents. This tutorial assumes that you are already familiar with the Windows environment and terms. To learn more about documents, maps, and jobs, refer to the appropriate section(s), as needed.



2. System Requirements

Ensure that your system meets the minimum requirements described below:

Hardware	Intel Pentium/Celeron processor at 1 Ghz or compatible CPU 512 MB of RAM At least 500 MB of available hard-disk space Internet Connection (Broadband is recommended)
Software	32 or 64 bit edition of Windows Vista/ 7/ 8 / 8.1 /Server 2008 / Server 2012

Notes:

- Installation requires Administrator privileges.
- Increasing processor speed and memory will significantly improve performance.
- Only Intel and compatible processors are supported.
- Audit Log and Reconciliation Viewer reports may require more memory.
- Disk space requirements are estimates and will vary depending on usage.
- Internet access is required.

Chapter II. Creating a Document

1. Documents

Documents can be subdivided into sections and delineated, for example, by headings. Headings can be considered to contain subsections (subheadings), as well as the information in the document.

In the **Document Manager/Data Link Manager** context, documents are composed of nodes, which are roughly equivalent to sections of documents. A node would correspond to section headings, section subheadings, tables, etc., in an actual document. Thus, nodes can be represented hierarchically. For example, one heading in a purchase order can have many order lines beneath it.

There are two types of nodes you will use when you create a document or mappings: Container Nodes and Detail Nodes. Container nodes contain no data, just other nodes, which can be container nodes and/or detail nodes; for example, segments, flat file records, and SQL tables are all container nodes. Detail nodes contain only data; for example, elements, flat file fields, and SQL fields (a.k.a. columns, attributes) are all detail nodes. You use these nodes to create a hierarchical document structure.

To perform a complete document conversion, you should have three documents:

- **Source document**: The original document that contains the data to be converted.
- **Target document**: The resulting document into which the data will be transferred.
- **Exchange document**: The intermediary document that tells the system into which section of the resulting document the data in the original document must be placed.

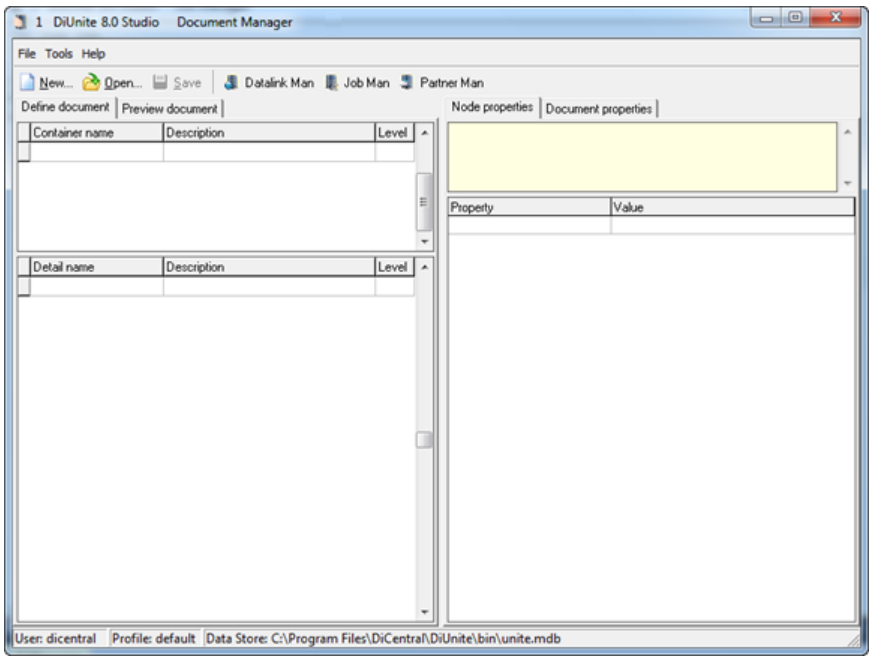
2. Starting Document Manager

To start Document Manager:

1. Click the Windows **Start** button. Next, point to **All Programs** > **DiCentral** > **DiUnite** and then click **Document Manager**. *The **Login** window is displayed.*

DiUnite 8.0 Studio Login
DiUnite 8.0 Studio Data Link Manager
User name:
dicentral
Password:
OK
Cancel
Options >>
©Copyright 2007 - 2011 DiCentral Corporation.
All rights reserved. This program is protected by US and international copyright laws as described in Help About.

2. Select the **User Name** field and enter your user name.
3. Select the **Password** field and enter your password.
4. Click the **OK** button. *The **Document Manager** window is displayed*.


1 DiUnite 8.0 Studio Document Manager
File Tools Help
New... Open... Save Datalink Man Job Man Partner Man
Define document Preview document
Node properties Document properties
Container name Description Level
Property Value
Detail name Description Level
User: dicentral Profile: default Data Store: C:\Program Files\DiCentral\DiUnite\bin\unite.mdb

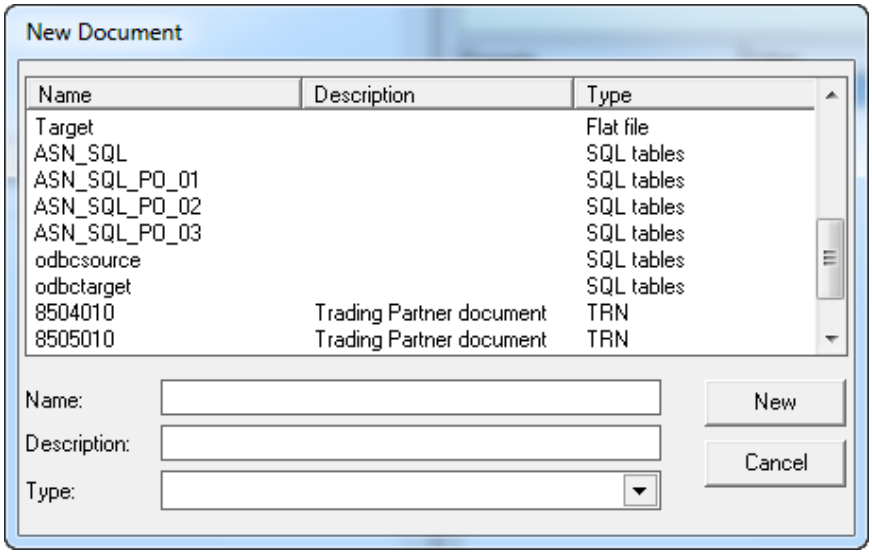
You add container nodes to the upper left work space in the **Document Manager** window. Once you create these nodes, you can build a detail node that belongs to the container node in the lower left pane.

3. Creating Source and Target Documents

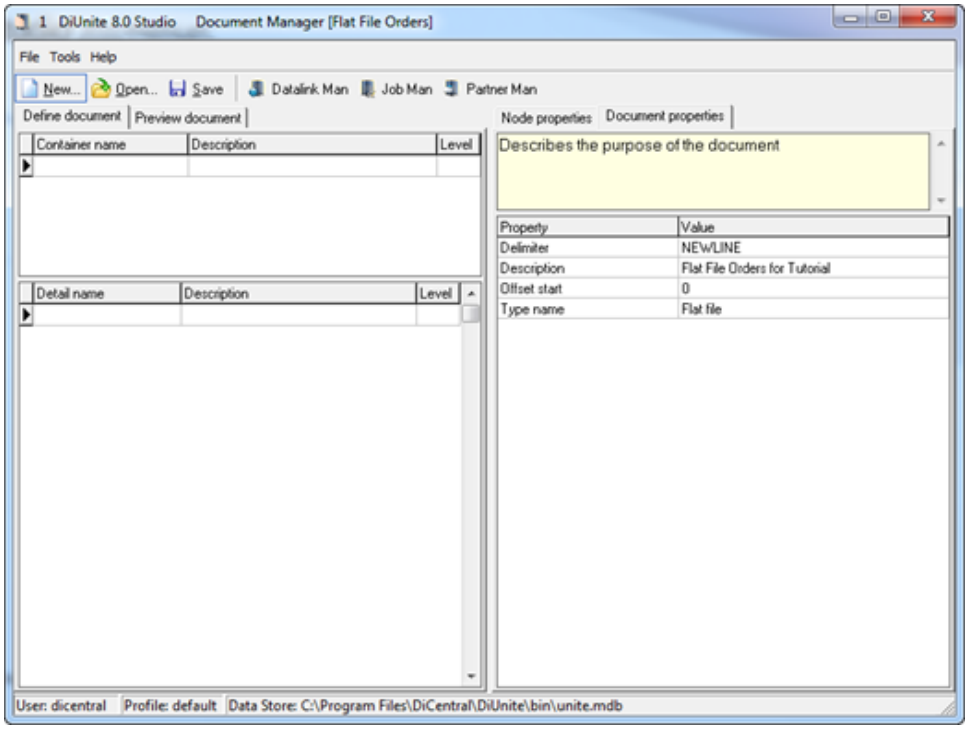
Creating a new document is an easy process. It is important to note that the procedures below will result in a very simple document. Nearly all of the documents you create will most likely be much more complicated. However, the principals in this section will always apply.

3.1 Creating a New Flat File Document

1. Click **New** in the **Document Manager Main** window. *The **New Document** window is displayed*.

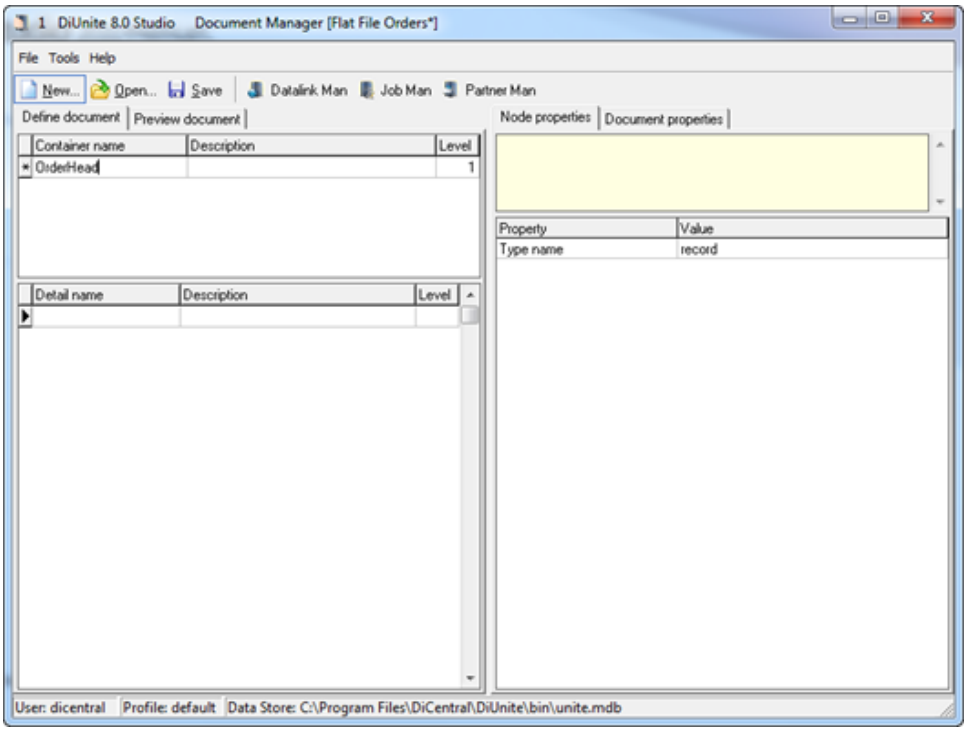


2. Select the **Name** field and enter **Source**.
3. Select the **Description** field and enter **Flat File Orders for Tutorial**.
4. Click the arrow for the **Type** dropdown list and select **Flat File**. Your options are:
   - Exchange (an intermediary document that has a suitable and common structure for easy mapping of source and target documents)
   - Flat File
   - SQL Tables
   - TRN (DiTranslator internal format)
5. Click **New** to create the document. The new document is ready to be edited in the **Document Manager** window. The name of the document appears in the **Title Bar**.


1 DiUnite 8.0 Studio Document Manager [Flat File Orders]
File Tools Help
New... Open... Save Datalink Man Job Man Partner Man
Define document Preview document
Container name Description Level
Detail name Description Level
Node properties Document properties
Describes the purpose of the document
Property Value
Delimiter NEWLINE
Description Flat File Orders for Tutorial
Offset start 0
Type name Flat file
User: dicentral Profile: default Data Store: C:\Program Files\DiCentral\DiUnite\bin\unite.mdb

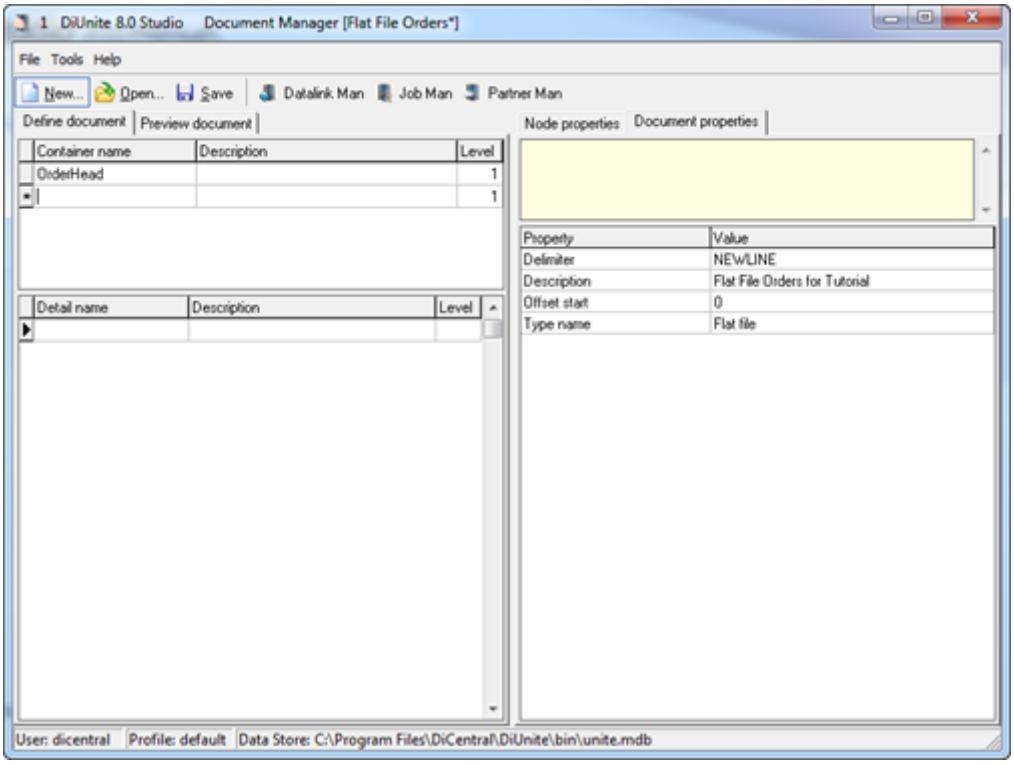
3.2 Adding Container Nodes to the New Document

1. Select the **Container Name** column and enter “OrderHead”, the name for this container node.
2. Select the **Description** column and enter “Order head virtual table”, the description for this container.
3. Select the **Level** column and enter the level of the hierarchy that this node resides in. In this case it is 1, which means it is the highest level of a hierarchy.
4. Select **Tab** for the Delimiter.

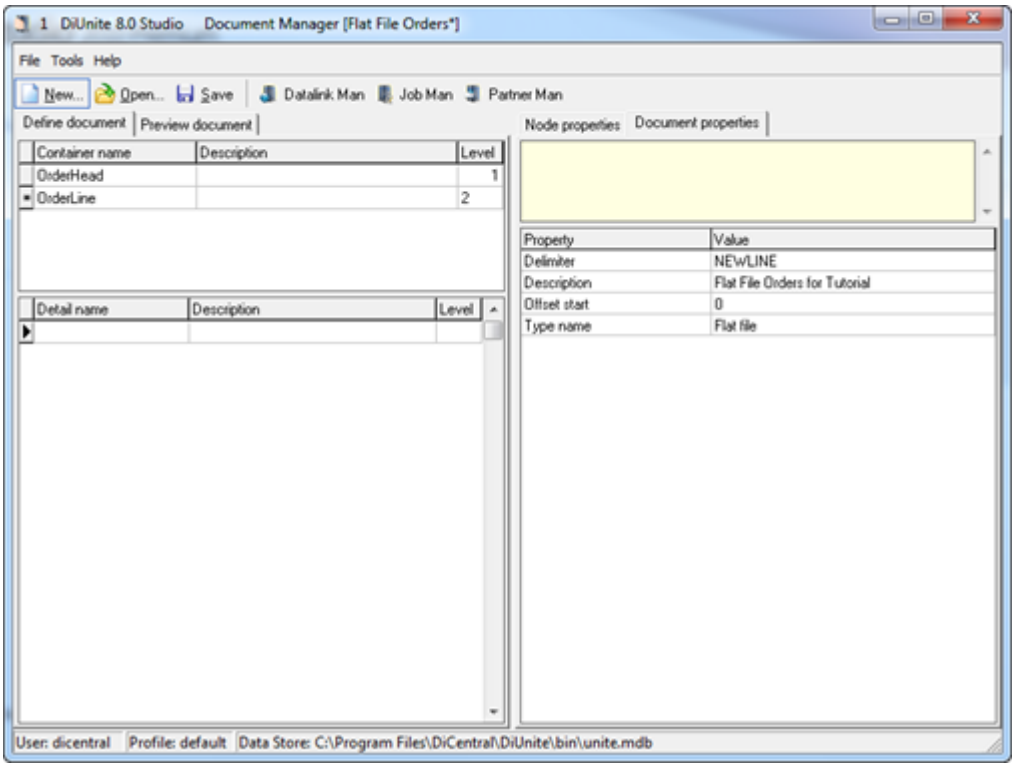

1 DiUnite 8.0 Studio Document Manager [Flat File Orders*]
File Tools Help
New... Open... Save Datalink Man Job Man Partner Man
Define document Preview document
Container name Description Level
OrderHead 1
Detail name Description Level
Node properties Document properties
Property Value
Type name record
User: dicentral Profile: default Data Store: C:\Program Files\DiCentral\DiUnite\bin\unite.mdb

5. Click the **Save** button. Your work will not be saved unless you click this button.

6. Press the down arrow key. *The node is added*.


1 DiUnite 8.0 Studio Document Manager [Flat File Orders*]
File Tools Help
New... Open... Save Datalink Man Job Man Partner Man
Define document Preview document
Node properties Document properties
Container name Description Level
OrderHead 1
1
Detail name Description Level
Property Value
Delimiter NEWLINE
Description Flat File Orders for Tutorial
Offset start 0
Type name Flat file
User: dicentral Profile: default Data Store: C:\Program Files\DiCentral\DiUnite\bin\unite.mdb

7. Select the **Container Name** column and enter “OrderLine”, the name for this current container node.
8. Select the **Description** column and enter “order line”, the description for the current container.
9. Select the **Level** column and enter “2”, the level of the hierarchy that this node resides in. This shows that this container node resides under the **Order Head** hierarchy.


1 DiUnite 8.0 Studio Document Manager [Flat File Orders*]
File Tools Help
New... Open... Save Datalink Man Job Man Partner Man
Define document Preview document
Node properties Document properties
Container name Description Level
OrderHead 1
OrderLine 2
Detail name Description Level
Property Value
Delimiter NEWLINE
Description Flat File Orders for Tutorial
Offset start 0
Type name Flat file
User: dicentral Profile: default Data Store: C:\Program Files\DiCentral\DiUnite\bin\unite.mdb

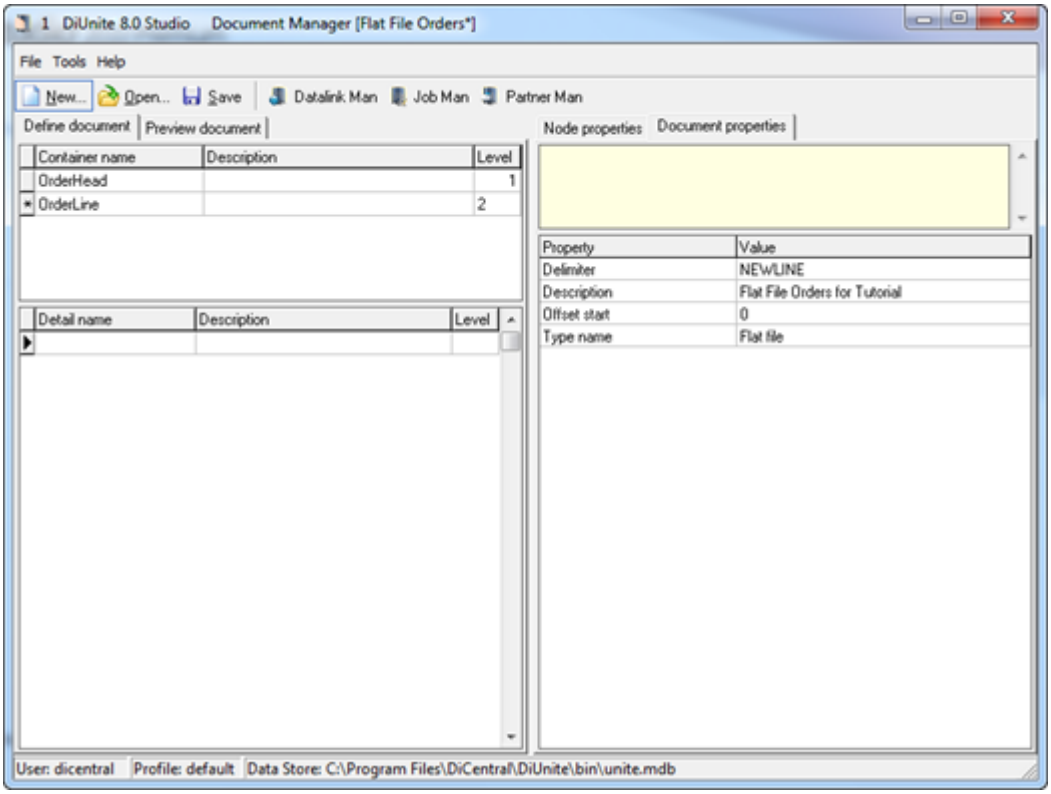
10. Click the **Save** button to save your work.

Note:
The table with the highest level in relational tables, or level 1, must be on top.

3.3 Adding Detail Node to the New Document

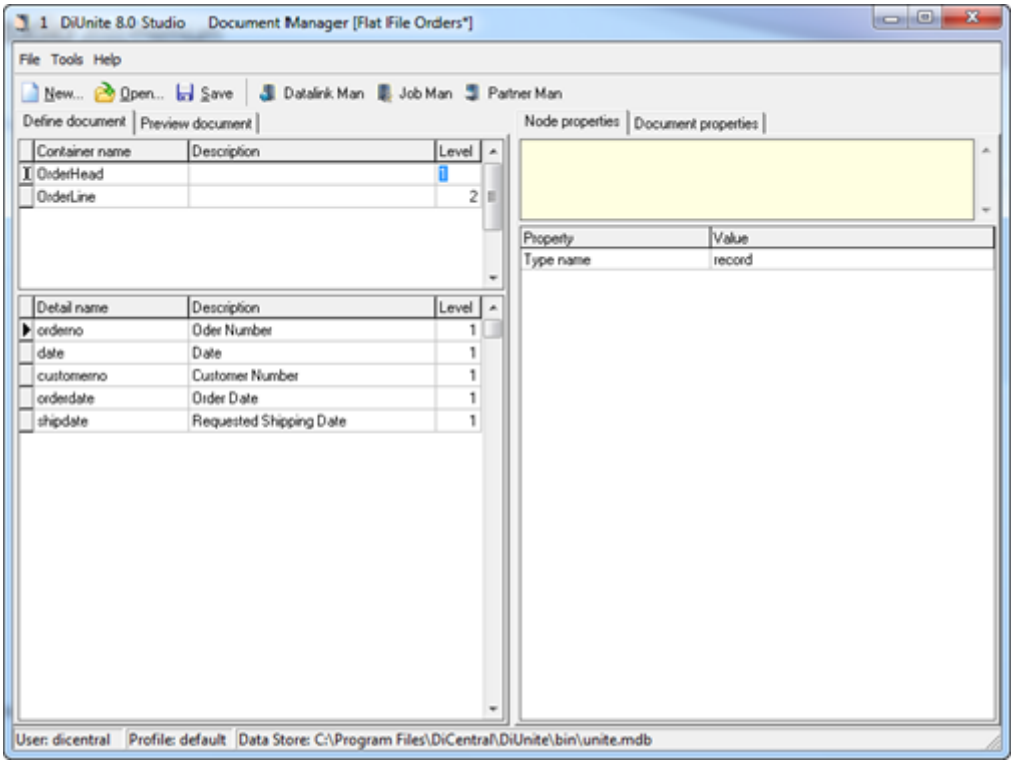
Start by adding detail nodes to the first container node.

1. Click the **OrderHead** container node to select it.
2. Select the **Detail Name** column and enter “orderno”, the name for the first detail node.
3. Select the **Description** column and enter “Order Number”, a description for the current container.
4. Select the **Level** column and enter “1”, the level of the hierarchy that this node resides in.
5. Select **Tab** for delimiter.

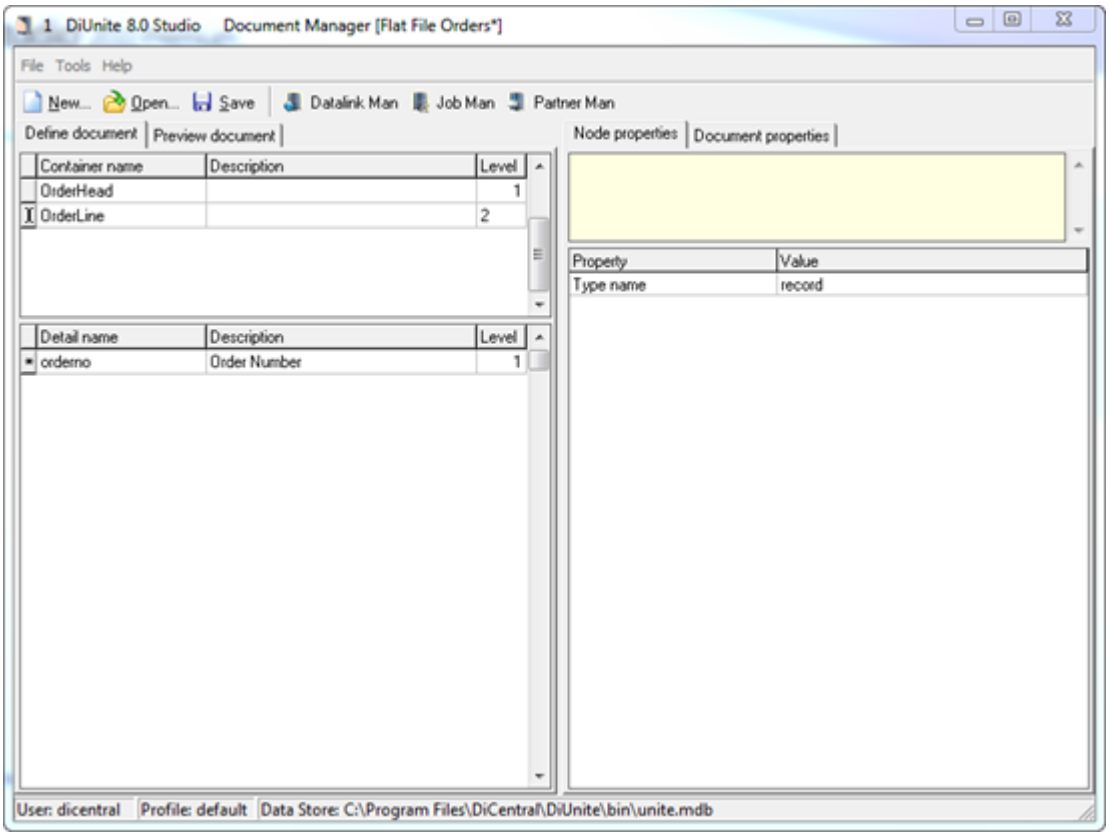

1 DiUnite 8.0 Studio Document Manager [Flat File Orders*]
File Tools Help
New... Open... Save Datalink Man Job Man Partner Man
Define document Preview document
Node properties Document properties
Container name Description Level
OrderHead 1
OrderLine 2
Detail name Description Level
Property Value
Delimiter NEWLINE
Description Flat File Orders for Tutorial
Offset start 0
Type name Flat file
User: dicentral Profile: default Data Store: C:\Program Files\DiCentral\DiUnite\bin\unite.mdb

6. Using the procedures above, create the following detail nodes in the OrderHead container node.

Name	Description	Level
date	Date	1
customerno	Customer Number	1
orderdate	Order Date	1
shipdate	Requested Shipping Date	1

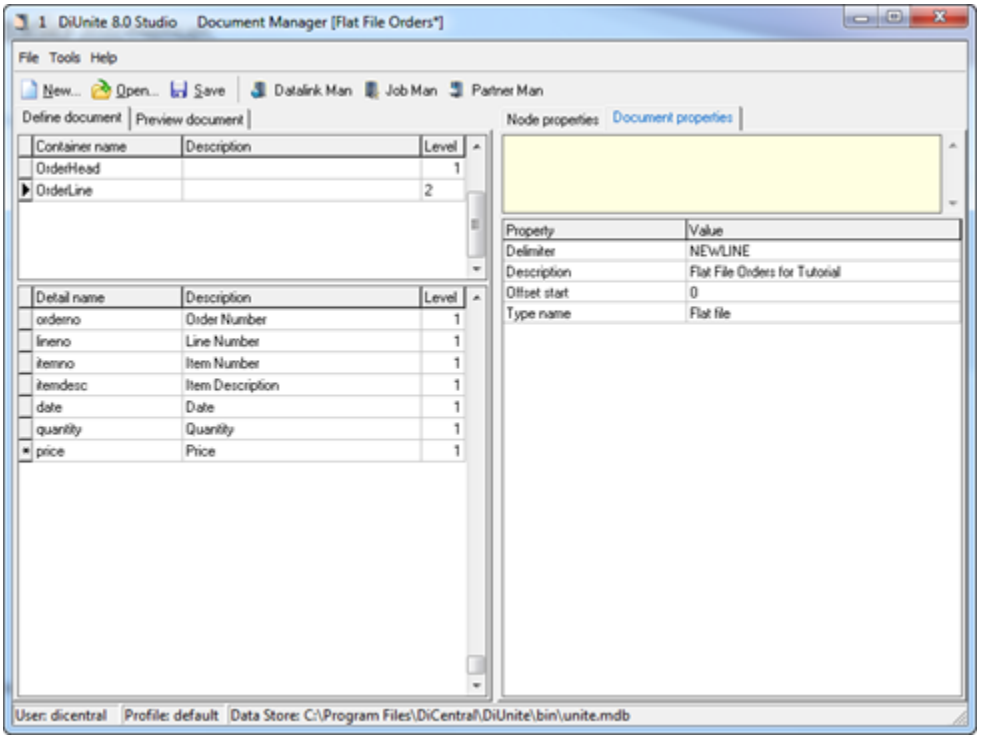

1 DiUnite 8.0 Studio Document Manager [Flat File Orders*]
File Tools Help
New... Open... Save Datalink Man Job Man Partner Man
Define document Preview document
Node properties Document properties
Container name Description Level
OrderHead 1
OrderLine 2
Property Value
Type name record
Detail name Description Level
orderno Oder Number 1
date Date 1
customerno Customer Number 1
orderdate Order Date 1
shipdate Requested Shipping Date 1
User: dicentral Profile: default Data Store: C:\Program Files\DiCentral\DiUnite\bin\unite.mdb

7. Click the **OrderLine** container node to select it.
8. Select the **Detail Name** column and enter “orderno”, the name for the first detail node.
9. Select the **Description** column and enter “Order Number”, the description for the current container.
10. Select the **Level** column and enter “1”, the level of the hierarchy that this node resides in.


1 DiUnite 8.0 Studio Document Manager [Flat File Orders*]
File Tools Help
New... Open... Save Datalink Man Job Man Partner Man
Define document Preview document
Node properties Document properties
Container name Description Level
OrderHead 1
OrderLine 2
Property Value
Type name record
Detail name Description Level
orderno Order Number 1
User: dicentral Profile: default Data Store: C:\Program Files\DiCentral\DiUnite\bin\unite.mdb

11. Press the down arrow key. *A node is added*. Using the procedures above, create the following detail nodes in the OrderLine container node.

Name	Description	Level
lineno	Line Number	1
itemno	Item Number	1
itemdesc	Item Description	1
date	Date	1
quantity	Quantity	1
price	Price	1

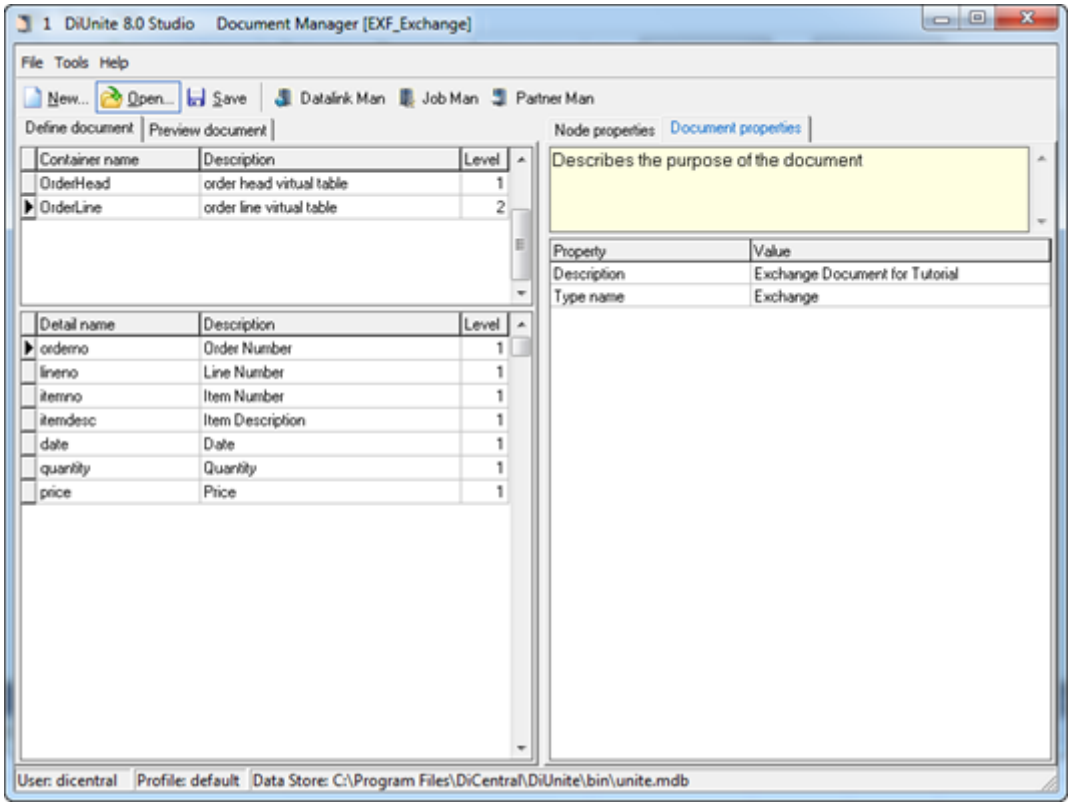

1 DiUnite 8.0 Studio Document Manager [Flat File Orders*]
File Tools Help
New... Open... Save Datalink Man Job Man Partner Man
Define document Preview document
Node properties Document properties
Container name Description Level
OrderHead 1
OrderLine 2
Detail name Description Level
orderno Order Number 1
lineno Line Number 1
itemno Item Number 1
itemdesc Item Description 1
date Date 1
quantity Quantity 1
price Price 1
Property Value
Delimiter NEWLINE
Description Flat File Orders for Tutorial
Offset start 0
Type name Flat file
User: dicentral Profile: default Data Store: C:\Program Files\DiCentral\DiUnite\bin\unite.mdb

12. Click the **Save** button to save your work.

3.4 Viewing the Document's Properties

Once you have created a document, you can view the properties connected to any node. To view a document's properties:

1. Click the **Document Properties** tab. *The **Document Manager** window is refreshed*.


1 DiUnite 8.0 Studio Document Manager [EXF_Exchange]
File Tools Help
New... Open... Save Datalink Man Job Man Partner Man
Define document Preview document
Node properties Document properties
Container name Description Level
OrderHead order head virtual table 1
OrderLine order line virtual table 2
Describes the purpose of the document
Property Value
Description Exchange Document for Tutorial
Type name Exchange
Detail name Description Level
orderno Order Number 1
lineno Line Number 1
itemno Item Number 1
itemdesc Item Description 1
date Date 1
quantity Quantity 1
price Price 1
User: dicentral Profile: default Data Store: C:\Program Files\DiCentral\DiUnite\bin\unite.mdb

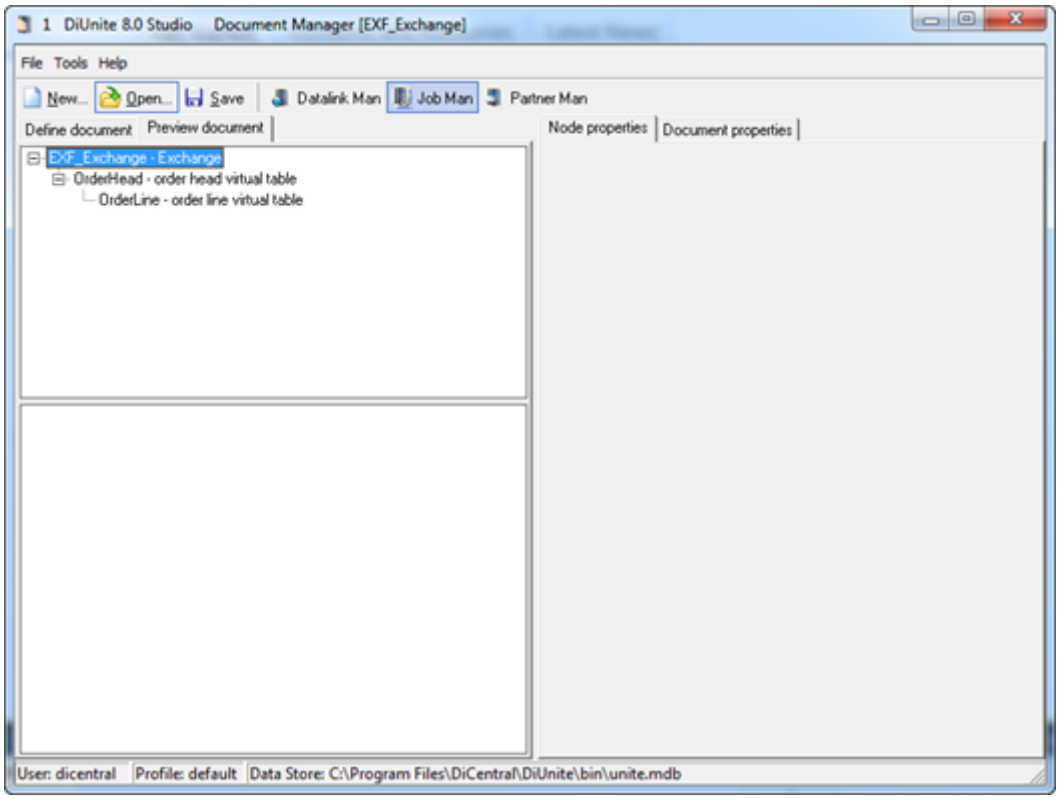
2. The description you entered when creating a document appears in the **Description** column. The type of document appears in the **Type name** row.

3.5 Previewing the Document

Once you have created a document, you can preview it. The preview version of the **Document Manager** window graphically displays the hierarchy you have created.

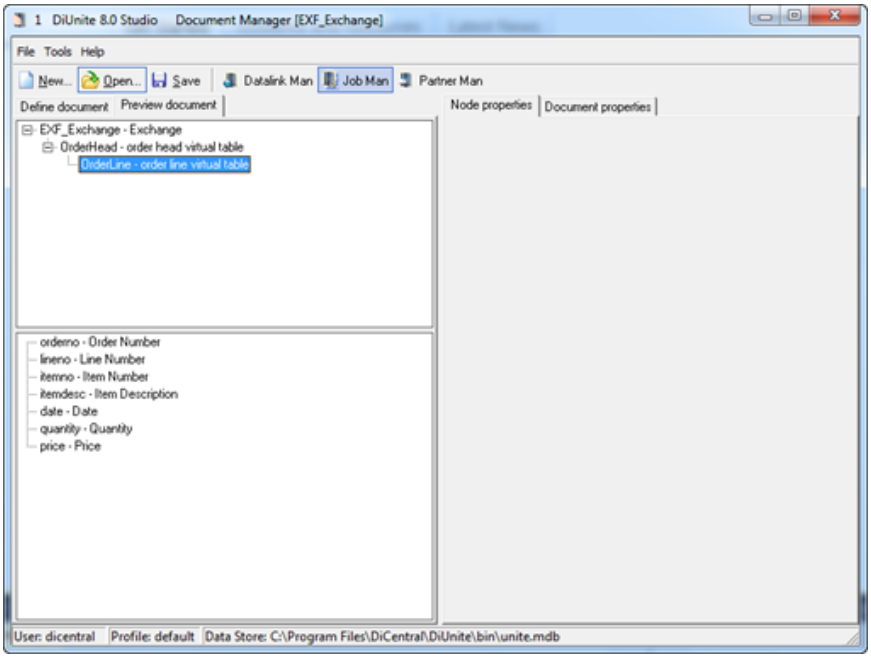
To preview a document

1. Click the **Preview** tab in the **Document Manager** window. *The **Preview** pane is displayed.*


1 DiUnite 8.0 Studio Document Manager [EXF_Exchange]
File Tools Help
New... Open... Save Datalink Man Job Man Partner Man
Define document Preview document
Node properties Document properties
EXF_Exchange - Exchange
OrderHead - order head virtual table
OrderLine - order line virtual table
User: dicentral Profile: default Data Store: C:\Program Files\DiCentral\DiUnite\bin\unite.mdb

2. To expand a branch of the document hierarchy, click the plus sign **(+)** next to the part of the hierarchy you want to expand.
3. To contract a branch of the hierarchy, click the minus sign **(-)** next to the part of the hierarchy you want to contract.

4. To drill down into a part of the hierarchy, click the part whose details you want to view.


1 DiUnite 8.0 Studio Document Manager [EXF_Exchange]
File Tools Help
New... Open... Save Datalink Man Job Man Partner Man
Define document Preview document
Node properties Document properties
EXF_Exchange - Exchange
OrderHead - order head virtual table
OrderLine - order line virtual table
orderno - Order Number
lineno - Line Number
itemno - Item Number
itemdesc - Item Description
date - Date
quantity - Quantity
price - Price
User: dicentral
Profile: default
Data Store: C:\Program Files\DiCentral\DiUnite\bin\unite.mdb

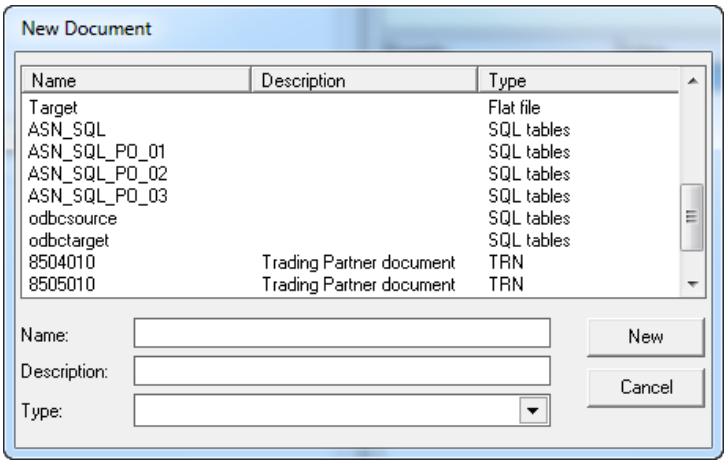
3.6 Validating the Document

Validating the document checks to see if the levels assigned to nodes are correct. For example, it checks if a node that belongs to level one, has been assigned to level two. You can only validate a document that is currently open.

To validate a document:

1. Select **Validate Document** from the **Tools** menu. **Document Manager** validates the document. When it is done, a dialog box appears informing you whether or not there are any errors in the document.
2. When you are finished with the information in the dialog box, click the **OK** button.
3. Select **Exit** from the **File** menu to exit **Document Manager**.

Note:
Repeat the same instructions to create an **Exchange** and **Target** document.
In the **Name** field, enter **Exchange** or **Target**.



4. Creating a Map

A mapping, or data link, is the linkage between one detail node in the source or target document and one detail node in the exchange document. You create this link by inserting a function in the appropriate source or target node. This can be done by dragging-and-dropping between the source or target node and the exchange node, or by selecting the node and inserting the function “manually” in the functions pane of the Data Link Manager window. Further manipulation on the linked value is done by adding more functions in the function list.

4.1 Starting Data Link Manager

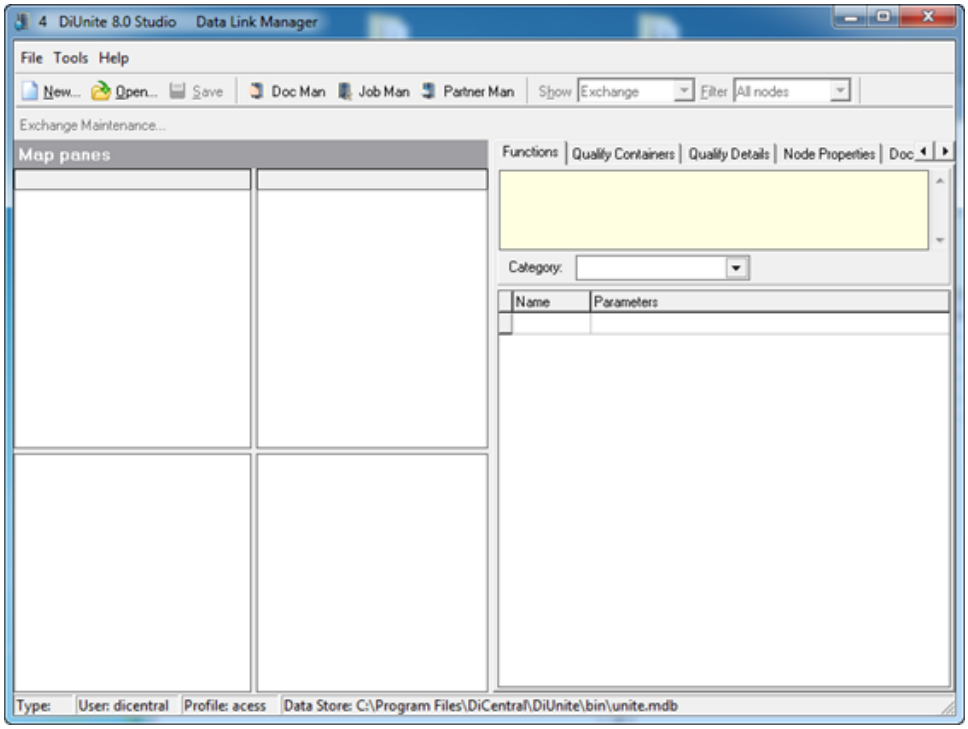
To start Document Manager:

1. Click the Windows **Start** button. Next, point to **All Programs** > **DiCentral** > **DiUnite** and then click **Data Link Manager**. *The **Login** window is displayed.*

DiUnite 8.0 Studio Login
DiUnite 8.0 Studio Data Link Manager
User name:
dicentral
Password:
OK
Cancel
Options >>
©Copyright 2007 - 2011 DiCentral Corporation.
All rights reserved. This program is protected by US and international copyright laws as described in Help About.

2. Select the **User Name** field and enter your user name.
3. Select the **Password** field and enter your password.

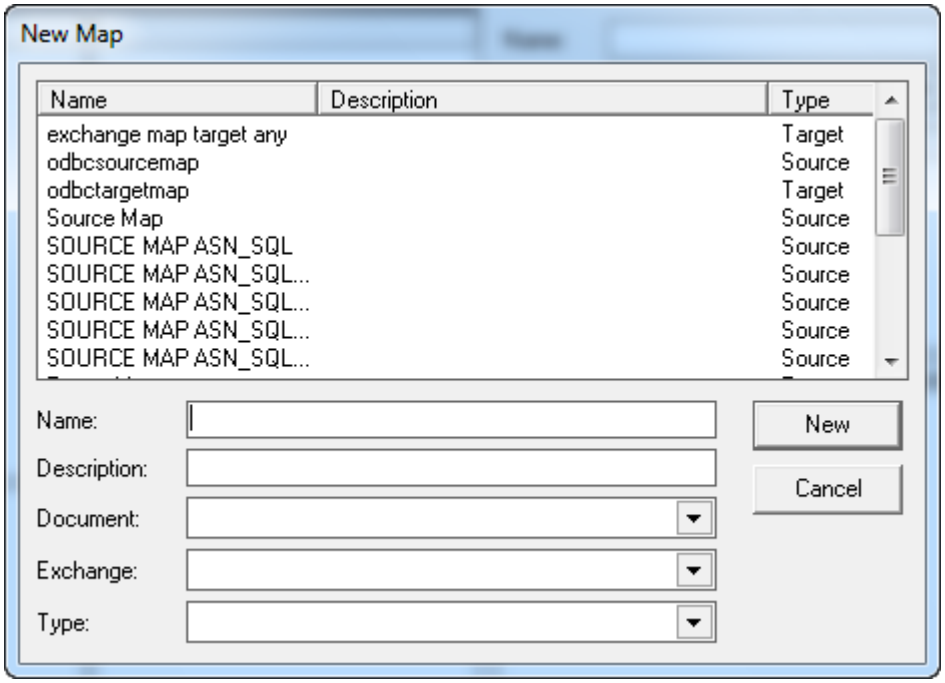
4. Click the **OK** button. *The **Data Link Manager** window is displayed*.


4 DiUnite 8.0 Studio Data Link Manager
File Tools Help
New... Open... Save Doc Man Job Man Partner Man Show Exchange Filter All nodes
Exchange Maintenance...
Map panes
Functions Qualify Containers Qualify Details Node Properties Doc
Category:
Name Parameters
Type: User: dicentral Profile: acess Data Store: C:\Program Files\DiCentral\DiUnite\bin\unite.mdb

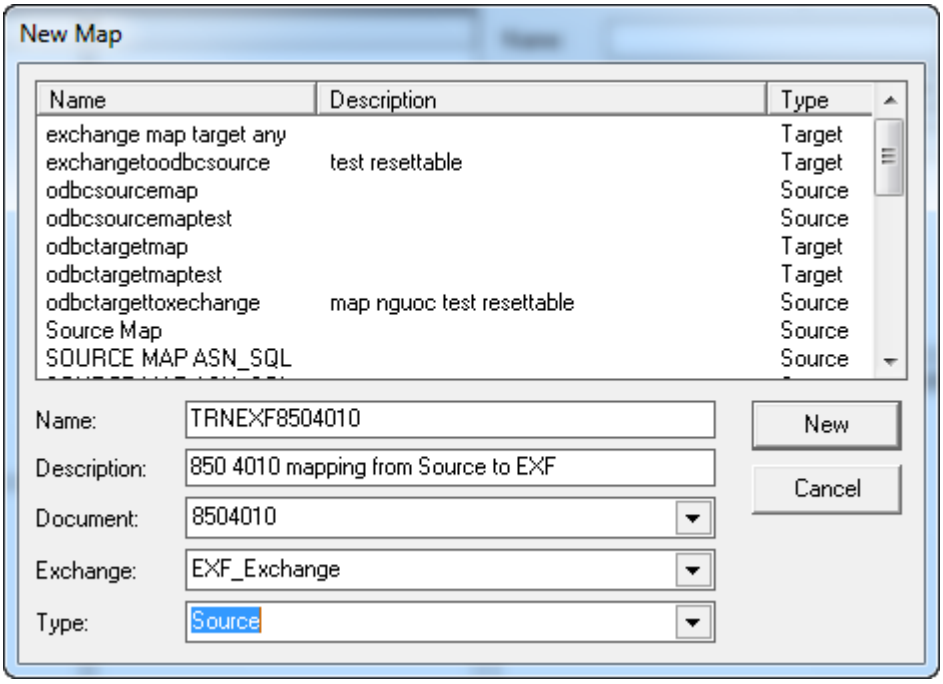
The map screen consists of two major areas. The left one is the Document area where you can drag-and-drop between the documents. The right one is the qualifying area.

4.2 Mapping from a Source Document to an Exchange Document

1. Click the **New** button. *The **New Map** screen is displayed*.

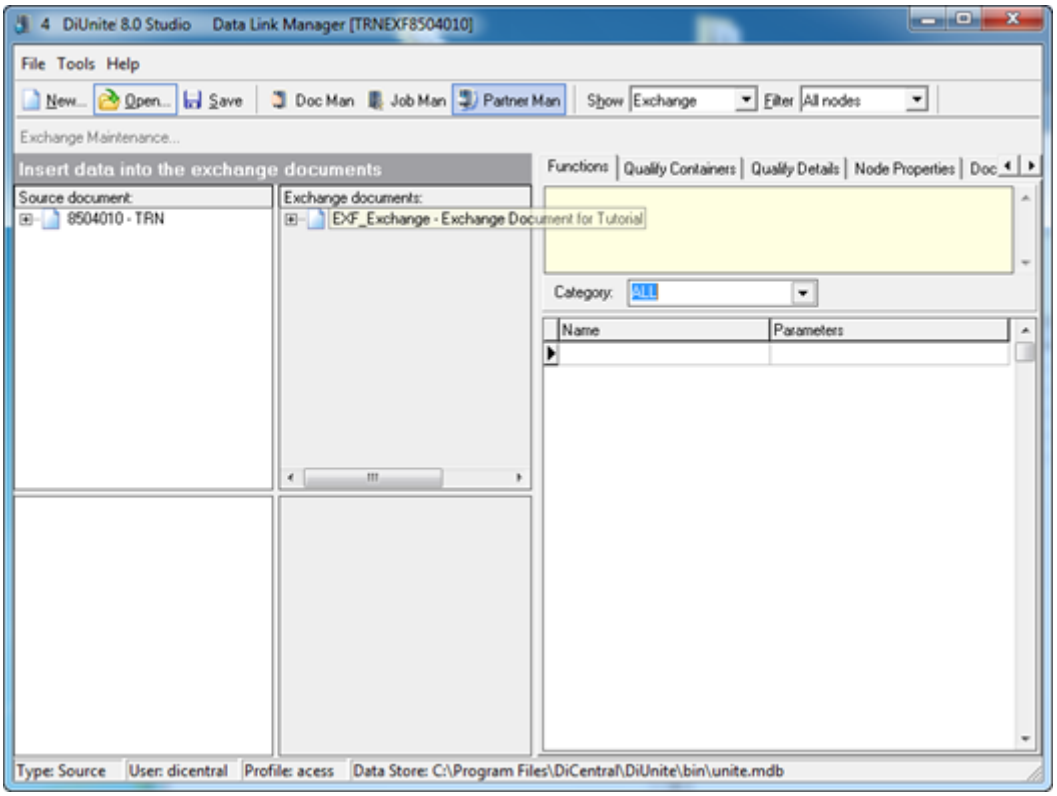


2. Select the **Name** field and enter “TRNEXD8504010”, the name of the new map.
3. Select the **Description** field and enter “850 4010 mapping from Source to EXD”, its description.
4. Click the arrow next to the **Document** field and select **8504010**, the document you will use as the source document.
5. Click the arrow next to the **Exchange** field and select **EXD_Exchange**, the document you will use as the exchange (EXD) document.
6. Click the arrow next to the **Type** field and select **Source**. This is the direction of this mapping. Your options are:
   - Source: in this mapping, the document will be a source document and the mapping will be from source to exchange.
   - Target: in this mapping, the document will be an exchange document and the mapping will be from exchange to target.



7. Click the **New** button.

The name of the TRN Source document appears in the upper left pane of the **Document Manager** window. The Exchange document (EXD_Exchange) appears in the right pane.

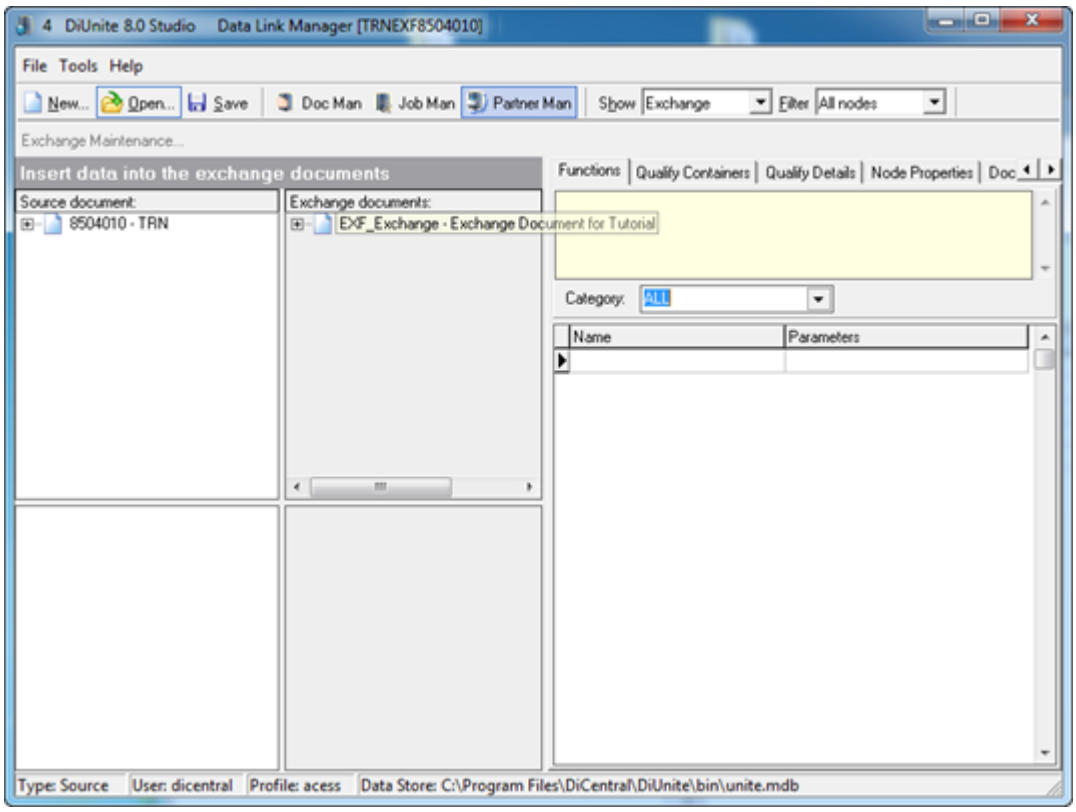

4 DiUnite 8.0 Studio Data Link Manager [TRNEXF8504010]
File Tools Help
New... Open... Save Doc Man Job Man Partner Man Show Exchange Filter All nodes
Exchange Maintenance...
Insert data into the exchange documents
Source document:
8504010 - TRN
Exchange documents:
EXF_Exchange - Exchange Document for Tutorial
Functions Quality Containers Quality Details Node Properties Doc
Category: ALL
Name Parameters
Type: Source User: dicentral Profile: acess Data Store: C:\Program Files\DiCentral\DiUnite\bin\unite.mdb

The **Source document** areas are white and the **Exchange document** areas are light grey. This means that no functions, properties, or qualifiers are assigned to the Exchange document.

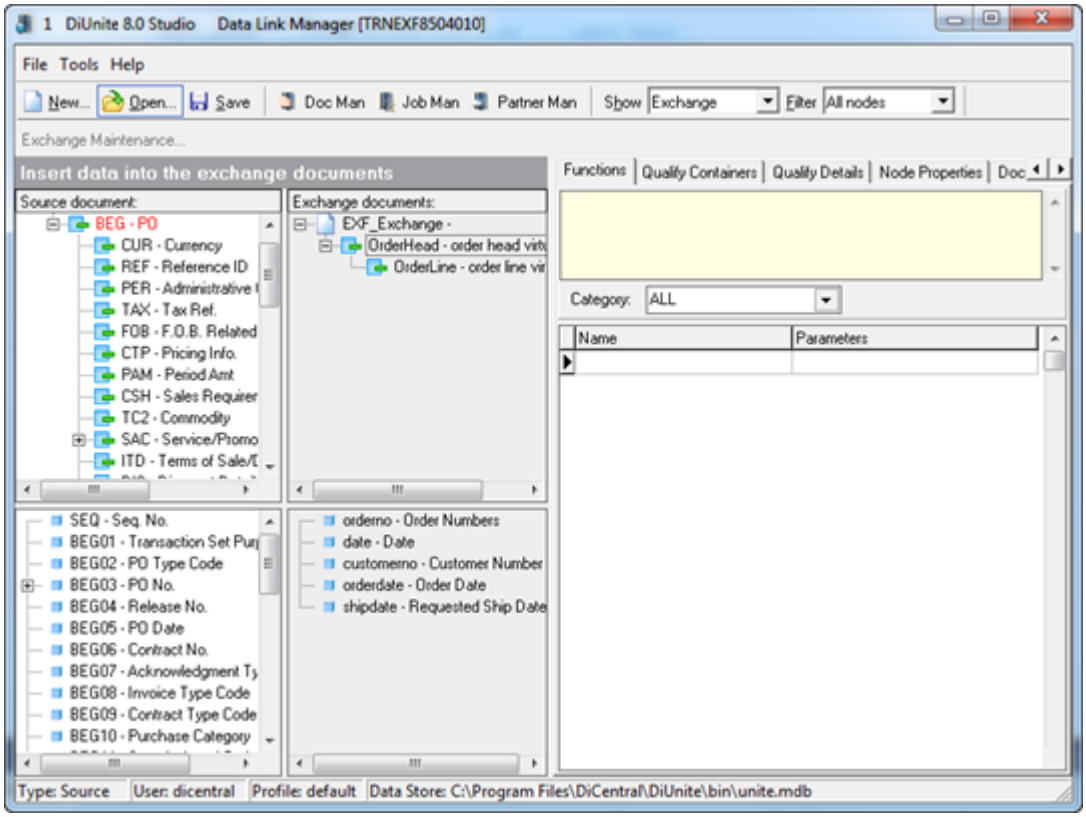
You will always create a mapping from **Left** to **Right**. When creating a Source to Exchange mapping, the **Source document** areas appear on the left and the mapping occurs to the **Exchange document** area which is on the right.

4.2.1 Setting a Function

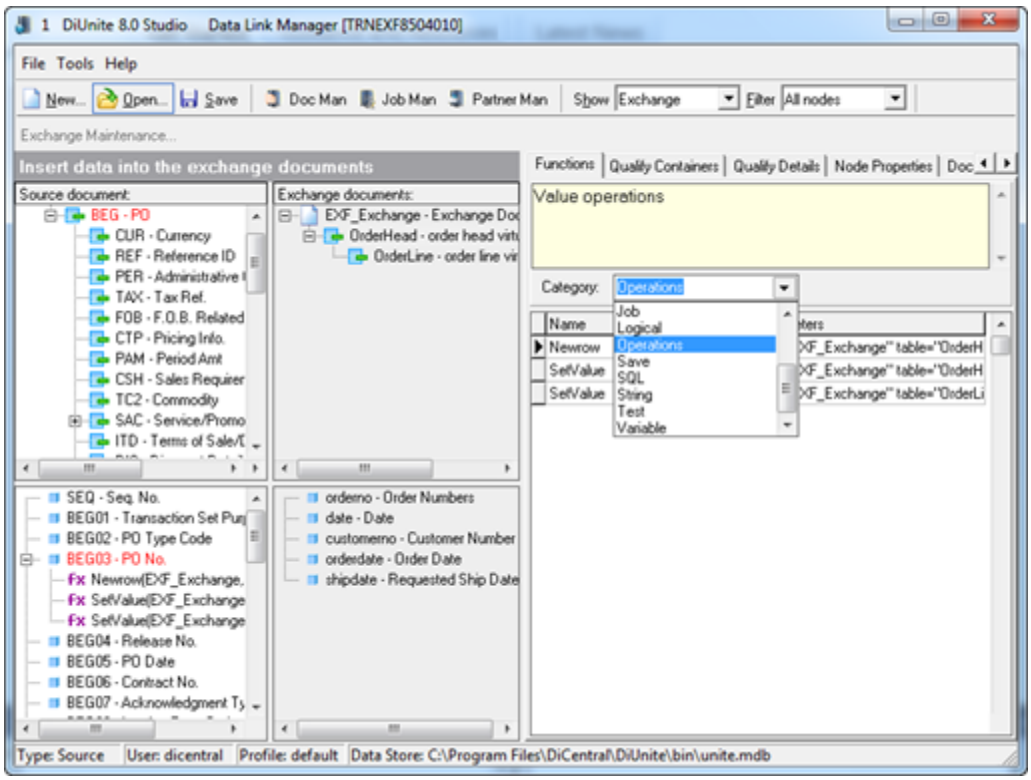
The first thing you need to decide is where to set the break points. Break points mark the location in the TRN structure where a new record will be inserted into the Exchange document. One break point is needed for each virtual table in the Exchange document. In this example, we need one for the Order Head and the other for the PO1 baseline item data level.


4 DiUnite 8.0 Studio Data Link Manager [TRNEXF8504010]
File Tools Help
New... Open... Save Doc Man Job Man Partner Man Show Exchange Filter All nodes
Exchange Maintenance...
Insert data into the exchange documents
Source document:
8504010 - TRN
Exchange documents:
EXF_Exchange - Exchange Document for Tutorial
Functions Qualify Containers Qualify Details Node Properties Doc
Category: ALL
Name Parameters
Type: Source User: dicentral Profile: acess Data Store: C:\Program Files\DiCentral\DiUnite\bin\unite.mdb

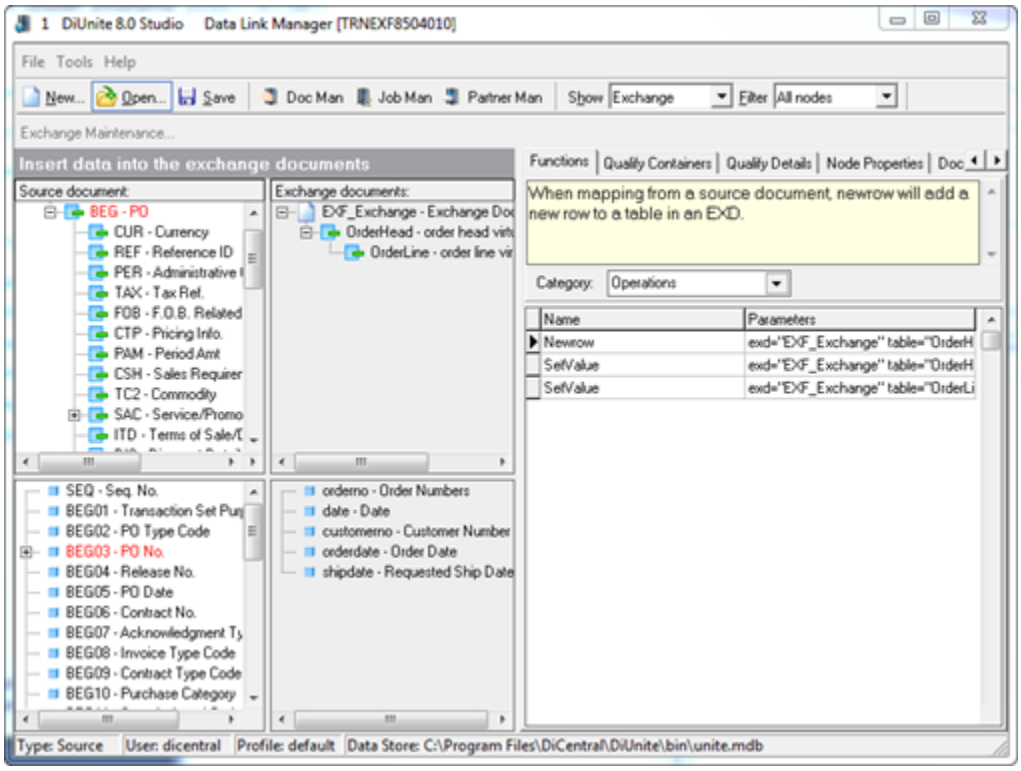
1. Click the **(+)** next to the Source document node(s) to expand the source document and expose all the nodes.
2. Click the **(+)** next to the Exchange document node(s) to expand the exchange document and expose all the nodes.


1 DiUnite 8.0 Studio Data Link Manager [TRNEXF8504010]
File Tools Help
New... Open... Save Doc Man Job Man Partner Man Show Exchange Filter All nodes
Exchange Maintenance...
Insert data into the exchange documents
Source document:
Exchange documents:
BEG - PO
CUR - Currency
REF - Reference ID
PER - Administrative
TAX - Tax Ref.
FOB - F.O.B. Related
CTP - Pricing Info.
PAM - Period Amt
CSH - Sales Requirer
TC2 - Commodity
SAC - Service/Promo
ITD - Terms of Sale/T
EXF_Exchange -
OrderHead - order head virt
OrderLine - order line vir
Functions Quality Containers Quality Details Node Properties Doc
Category: ALL
Name Parameters
SEQ - Seq. No.
BEG01 - Transaction Set Pur
BEG02 - PO Type Code
BEG03 - PO No.
BEG04 - Release No.
BEG05 - PO Date
BEG06 - Contract No.
BEG07 - Acknowledgment Ty
BEG08 - Invoice Type Code
BEG09 - Contract Type Code
BEG10 - Purchase Category
orderno - Order Numbers
date - Date
customerno - Customer Number
orderdate - Order Date
shipdate - Requested Ship Date
Type: Source User: dicentral Profile: default Data Store: C:\Program Files\DiCentral\DiUnite\bin\unite.mdb

3. Click the **BEG-PO** container node in the source document to select that container node.
4. Click the **OrderHead** container node in the exchange document to select that container node.
5. Click the **BEG03-PO No.** detail node in the source document to select that detail node.

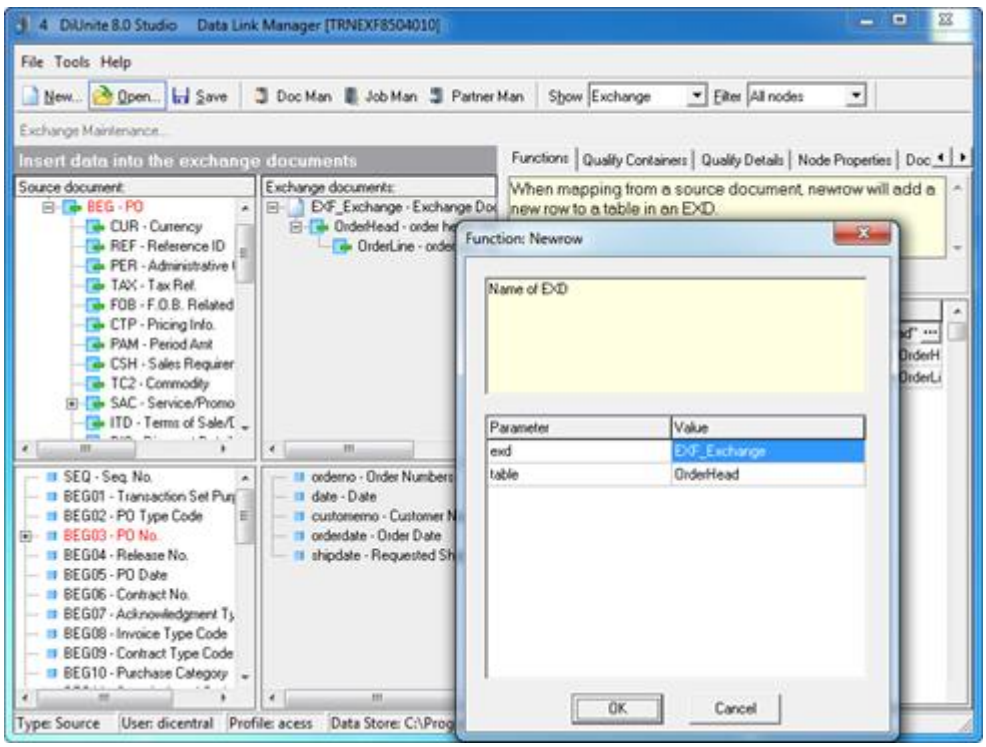


6. Click the blank box in the **Name** column on the right side of the **Data Link Manager** window and select **Newrow** from the dropdown list that appears.


1 DiUnite 8.0 Studio Data Link Manager [TRNEXF8504010]
File Tools Help
New... Open... Save Doc Man Job Man Partner Man Show Exchange Filter All nodes
Exchange Maintenance...
Insert data into the exchange documents
Source document:
Exchange documents:
Functions Quality Containers Quality Details Node Properties Doc
When mapping from a source document, newrow will add a new row to a table in an EXD.
Category: Operations
Name Parameters
Newrow exd="EXF_Exchange" table="OrderH
SetValue exd="EXF_Exchange" table="OrderH
SetValue exd="EXF_Exchange" table="OrderLi
Type: Source User: dicentral Profile: default Data Store: C:\Program Files\DiCentral\DiUnite\bin\unite.mdb

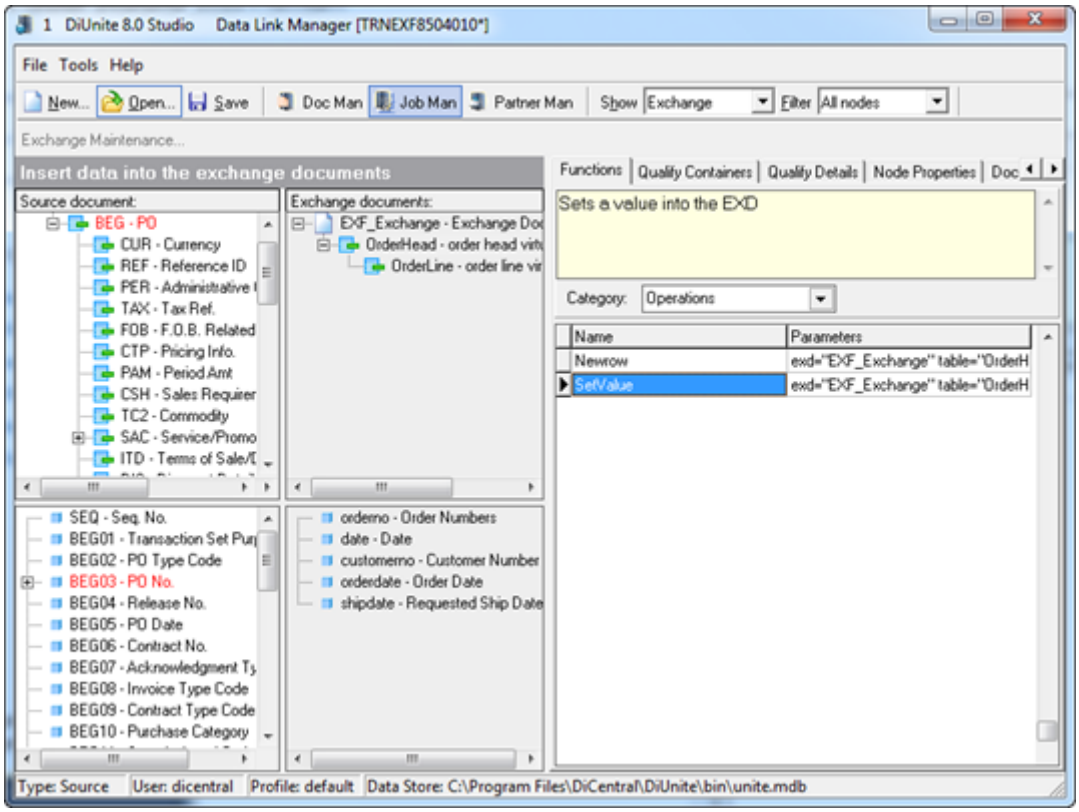
7. Select the parameters field and enter **exd=”EXD_Exchange” table=”OrderHead”** in the Functions help window and paste it in the blank field of the Parameters column.

This determines in which Exchange document you want to create a new row called “Newrow” and in which table.



Now, map the Order number from the source document to the Order number in the exchange document.

8. Click the **OrderHead** container node in the exchange document.
9. Click the **BEG03 – PO No.** detail node in the source document and drag-and-drop it over the orderno – order number detail node in the Exchange document. A new function called SetValue appears in the function list, with new parameters.

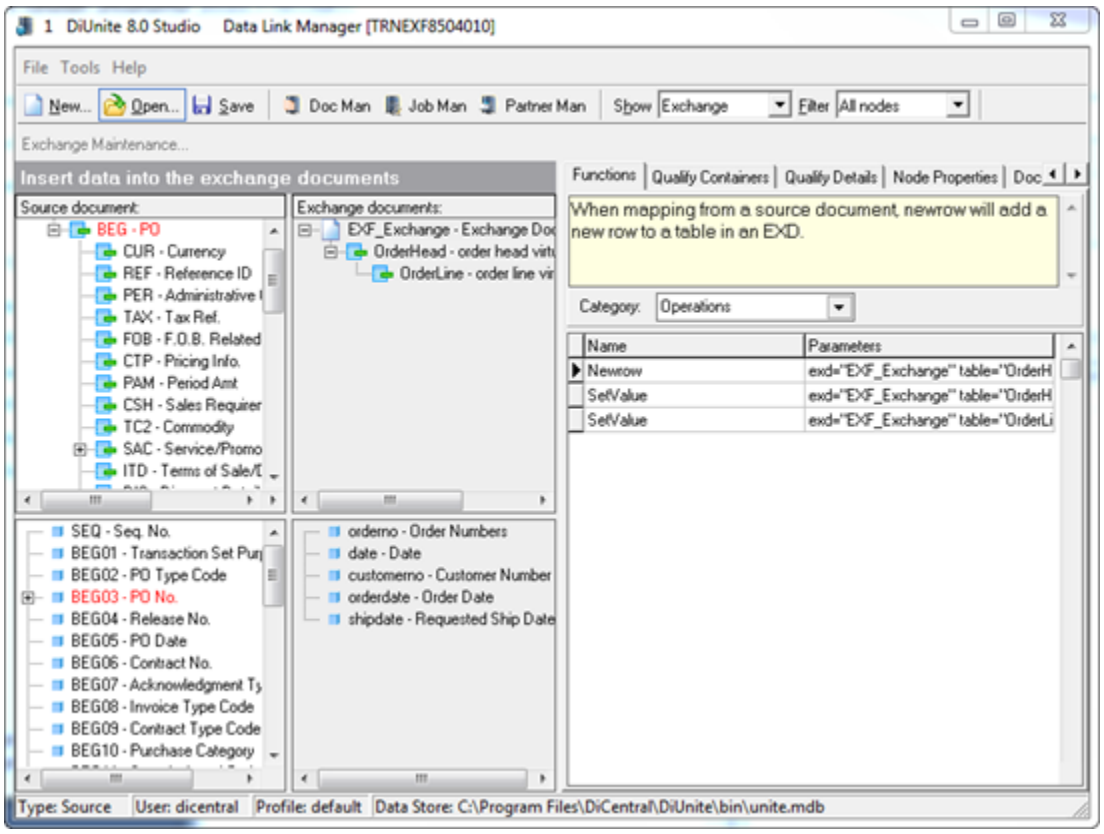

1 DiUnite 8.0 Studio Data Link Manager [TRNEXF8504010*]
File Tools Help
New... Open... Save Doc Man Job Man Partner Man Show Exchange Filter All nodes
Exchange Maintenance...
Insert data into the exchange documents
Source document:
BEG - PO
CUR - Currency
REF - Reference ID
PER - Administrative
TAX - Tax Ref.
FOB - F.O.B. Related
CTP - Pricing Info.
PAM - Period Amt
CSH - Sales Requirer
TC2 - Commodity
SAC - Service/Promo
ITD - Terms of Sale/T
Exchange documents:
EXF_Exchange - Exchange Doc
OrderHead - order head virt
OrderLine - order line vir
Functions Qualify Containers Qualify Details Node Properties Doc
Sets a value into the EXD
Category: Operations
Name Parameters
Newrow exd="EXF_Exchange" table="OrderH
SetValue exd="EXF_Exchange" table="OrderH
SEQ - Seq. No.
BEG01 - Transaction Set Pur
BEG02 - PO Type Code
BEG03 - PO No.
BEG04 - Release No.
BEG05 - PO Date
BEG06 - Contract No.
BEG07 - Acknowledgment Ty
BEG08 - Invoice Type Code
BEG09 - Contract Type Code
BEG10 - Purchase Category
orderno - Order Numbers
date - Date
customerno - Customer Number
orderdate - Order Date
shipdate - Requested Ship Date
Type: Source User: dicentral Profile: default Data Store: C:\Program Files\DiCentral\DiUnite\bin\unite.mdb

In this case the order number will also be a key field in the OrderLine table. This means that the order number plus the line number is the unique key for each order line. You then need to map the order number to the order line, as well.

Now you will change the focus on the Exchange document over to the OrderLine table and see the order line fields in the working area below.

10. Click the **OrderLine** container node in the exchange document.

11. Click the **BEG03 – PO No.** detail node in the source document and drag it over and drop it on the orderno – order number in the exchange document. As a result, the BEG03 – PO No. data element now has 3 functions mapped to it.


1 DiUnite 8.0 Studio Data Link Manager [TRNEXF8504010]
File Tools Help
New... Open... Save Doc Man Job Man Partner Man Show Exchange Filter All nodes
Exchange Maintenance...
Insert data into the exchange documents
Source document:
BEG - PO
CUR - Currency
REF - Reference ID
PER - Administrative
TAX - Tax Ref.
FOB - F.O.B. Related
CTP - Pricing Info.
PAM - Period Amt
CSH - Sales Requirer
TC2 - Commodity
SAC - Service/Promo
ITD - Terms of Sale/
SEQ - Seq. No.
BEG01 - Transaction Set Pur
BEG02 - PO Type Code
BEG03 - PO No.
BEG04 - Release No.
BEG05 - PO Date
BEG06 - Contract No.
BEG07 - Acknowledgment Ty
BEG08 - Invoice Type Code
BEG09 - Contract Type Code
BEG10 - Purchase Category
Exchange documents:
EXF_Exchange - Exchange Doc
OrderHead - order head virt
OrderLine - order line vir
orderno - Order Numbers
date - Date
customerno - Customer Number
orderdate - Order Date
shipdate - Requested Ship Date
Functions Qualify Containers Qualify Details Node Properties Doc
When mapping from a source document, newrow will add a new row to a table in an EXD.
Category: Operations
Name Parameters
Newrow exd="EXF_Exchange" table="OrderH
SetValue exd="EXF_Exchange" table="OrderH
SetValue exd="EXF_Exchange" table="OrderLi
Type: Source User: dicentral Profile: default Data Store: C:\Program Files\DiCentral\DiUnite\bin\unite.mdb

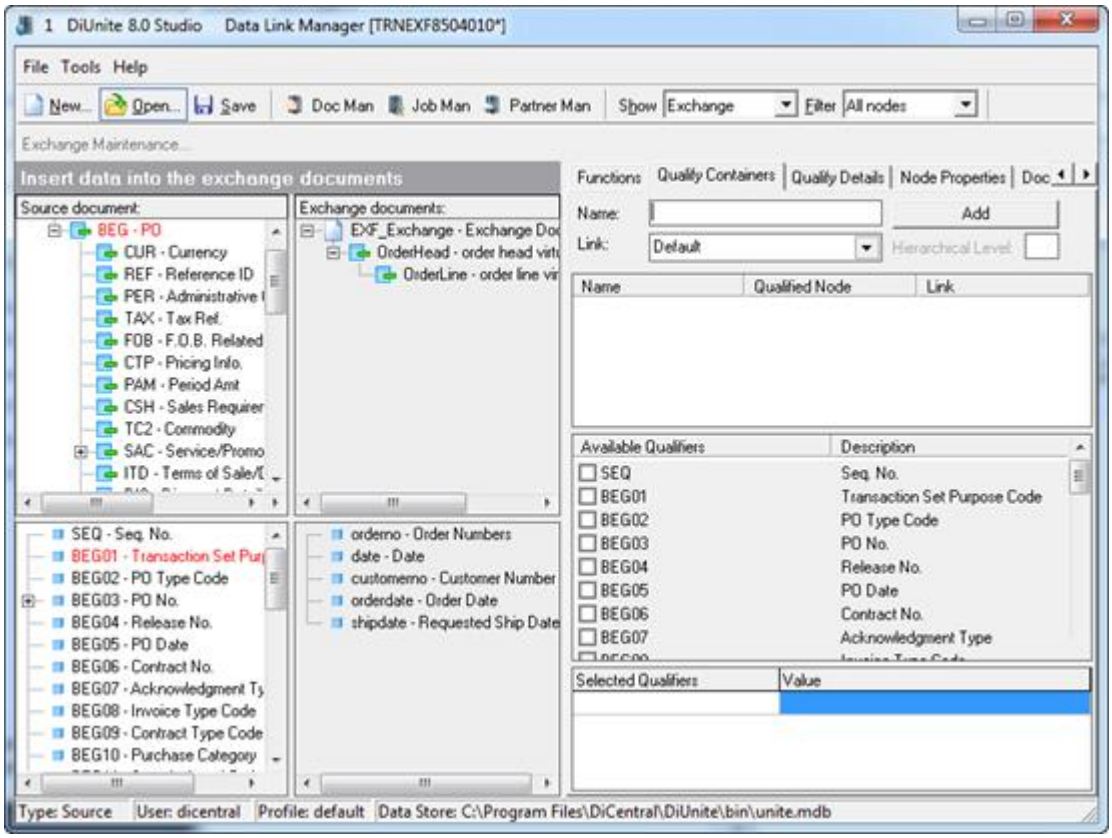
The **New Row** function means every time the conversion finds a new BG03 data element, it will start a new record in the Order Head table. The first **SetValue** function maps the Order number from this field over to the EXD Order number on the Header Table. The second **SetValue** function creates the same map and puts the order number into the OrderLine record.

12. Click the **Save** button to save the map.

4.2.2 Qualifying an Element

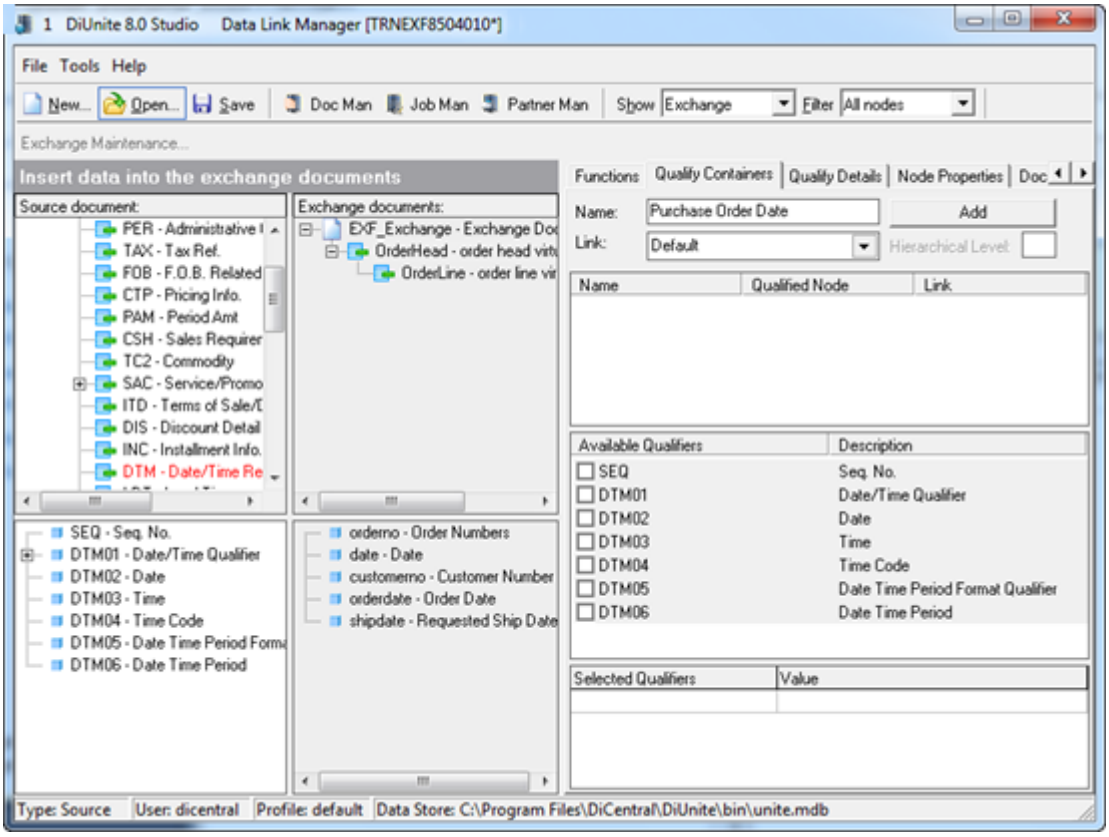
In the exchange document, the **OrderHead** container node contains two detail nodes: one titled Order Date and one titled Requested Shipping Date. To map these two fields, you must qualify the **DTM (Date/Time) Segment** on the source document.

1. Click the **OrderHead** container node in the exchange document.
2. Click the **DTM** container node in the source document.
3. Click the **Container Qualifiers** tab in the upper right pane. *The **Data Link Manager** window is refreshed*.



4. Select the **Name** field and enter **Purchase Order Date**.

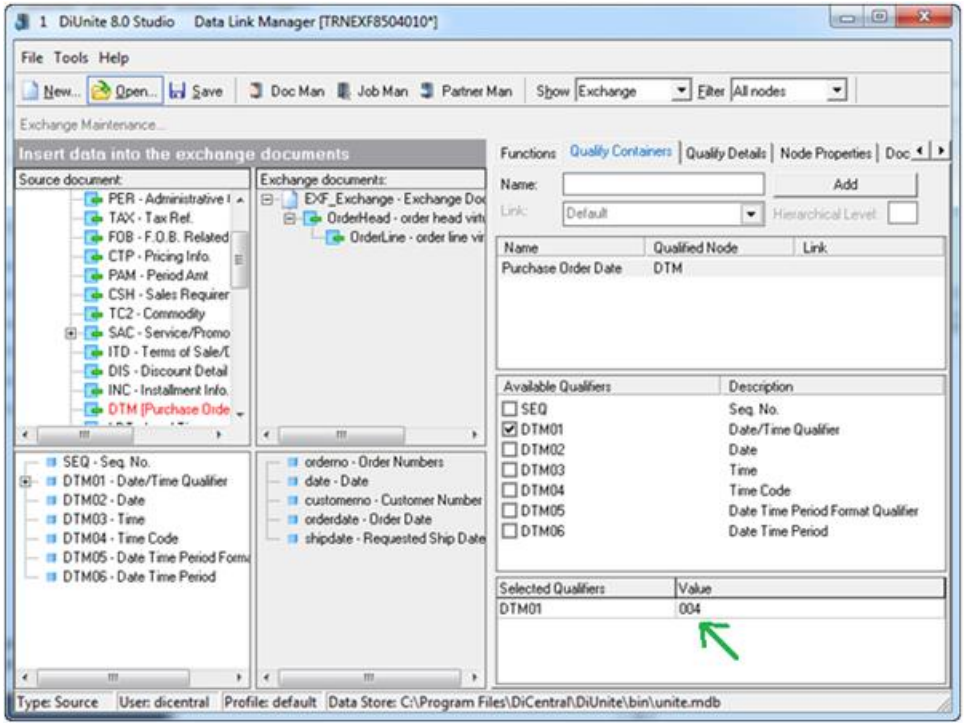
5. Click **Add Qualifier**. The name of the qualifier is added to the **Available Qualifier** list.



Now, select the field that qualifies the segment.

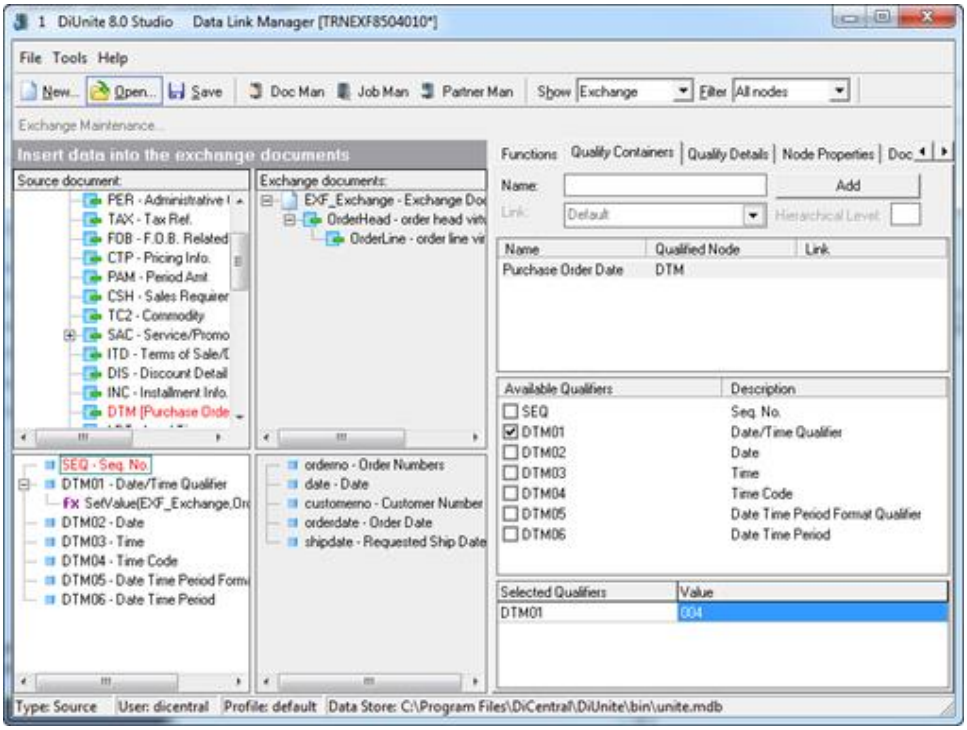
6. Click the **DTM01 Date/Time Qualifier** checkbox to check it. The data field will now appear in the window below.
7. Select the **Value** field and enter 004.

Qualifying information for DTM01 appears in the working area, in the Qualifying Node column and 004 in the Value column. In the Source document window, Purchase Date appears between the segment name (DTM) and the description (Date/Time Ref).


1 DiUnite 8.0 Studio Data Link Manager [TRNEXF8504010*]
File Tools Help
New... Open... Save Doc Man Job Man Partner Man Show Exchange Filter All nodes
Exchange Maintenance...
Insert data into the exchange documents
Source document:
Exchange documents:
Functions Quality Containers Quality Details Node Properties Doc
Name: Add
Link: Default Hierarchical Level
Name Qualified Node Link
Purchase Order Date DTM
Available Qualifiers Description
SEQ Seq. No.
DTM01 Date/Time Qualifier
DTM02 Date
DTM03 Time
DTM04 Time Code
DTM05 Date Time Period Format Qualifier
DTM06 Date Time Period
Selected Qualifiers Value
DTM01 004
SEQ - Seq. No.
DTM01 - Date/Time Qualifier
DTM02 - Date
DTM03 - Time
DTM04 - Time Code
DTM05 - Date Time Period Form
DTM06 - Date Time Period
orderno - Order Numbers
date - Date
customerno - Customer Number
orderdate - Order Date
shipdate - Requested Ship Date
Type: Source User: dicentral Profile: default Data Store: C:\Program Files\DiCentral\DiUnite\bin\unite.mdb

In the Source document detail node area, you can now map the purchase date from the source document to the exchange document.

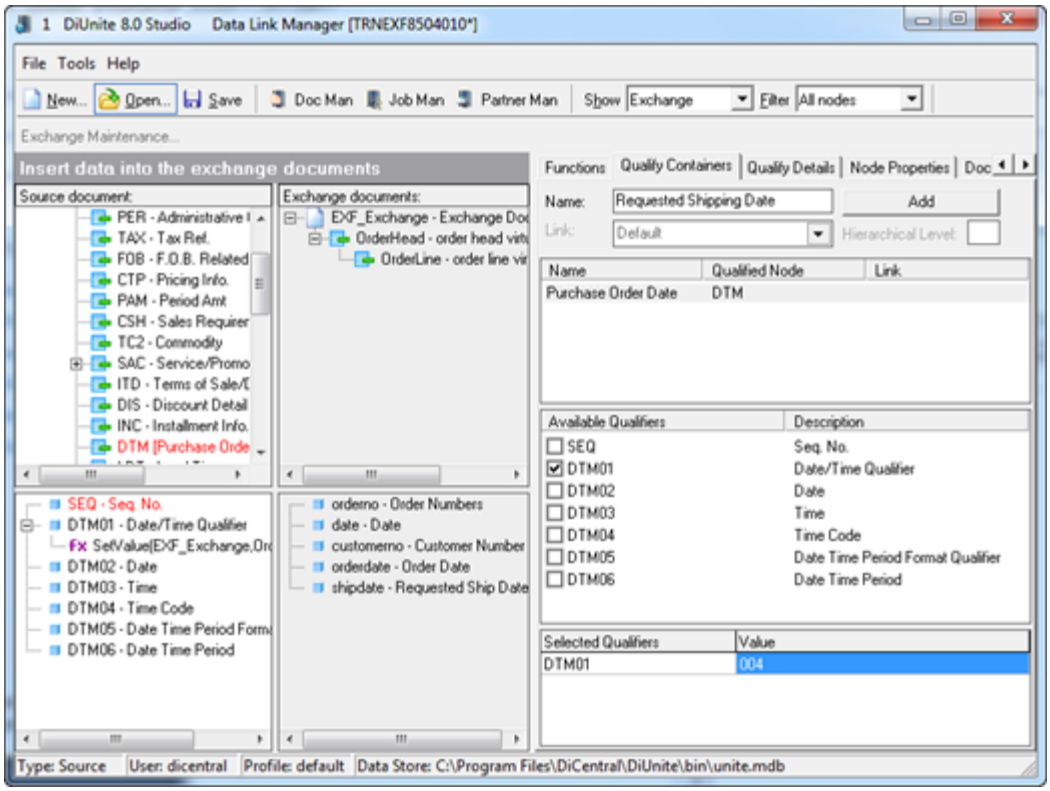
8. Click the **DTM02** detail node in the source document and drag-and-drop it over the orderdate detail node in the exchange document.


1 DiUnite 8.0 Studio Data Link Manager [TRNEXF8504010*]
File Tools Help
New... Open... Save Doc Man Job Man Partner Man Show Exchange Filter All nodes
Exchange Maintenance...
Insert data into the exchange documents
Source document:
Exchange documents:
Functions Quality Containers Quality Details Node Properties Doc
Name: Add
Link: Default Hierarchical Level
Name Qualified Node Link
Purchase Order Date DTM
Available Qualifiers Description
SEQ Seq. No.
DTM01 Date/Time Qualifier
DTM02 Date
DTM03 Time
DTM04 Time Code
DTM05 Date Time Period Format Qualifier
DTM06 Date Time Period
Selected Qualifiers Value
DTM01 004
SEQ - Seq. No.
DTM01 - Date/Time Qualifier
DTM02 - Date
DTM03 - Time
DTM04 - Time Code
DTM05 - Date Time Period Form
DTM06 - Date Time Period
orderno - Order Numbers
date - Date
customerno - Customer Number
orderdate - Order Date
shipdate - Requested Ship Date
Type: Source User: dicentral Profile: default Data Store: C:\Program Files\DiCentral\DiUnite\bin\unite.mdb

Next, you will add **Requested Shipping Date**. For this, you need to map a new date to the **OrderHead** table.

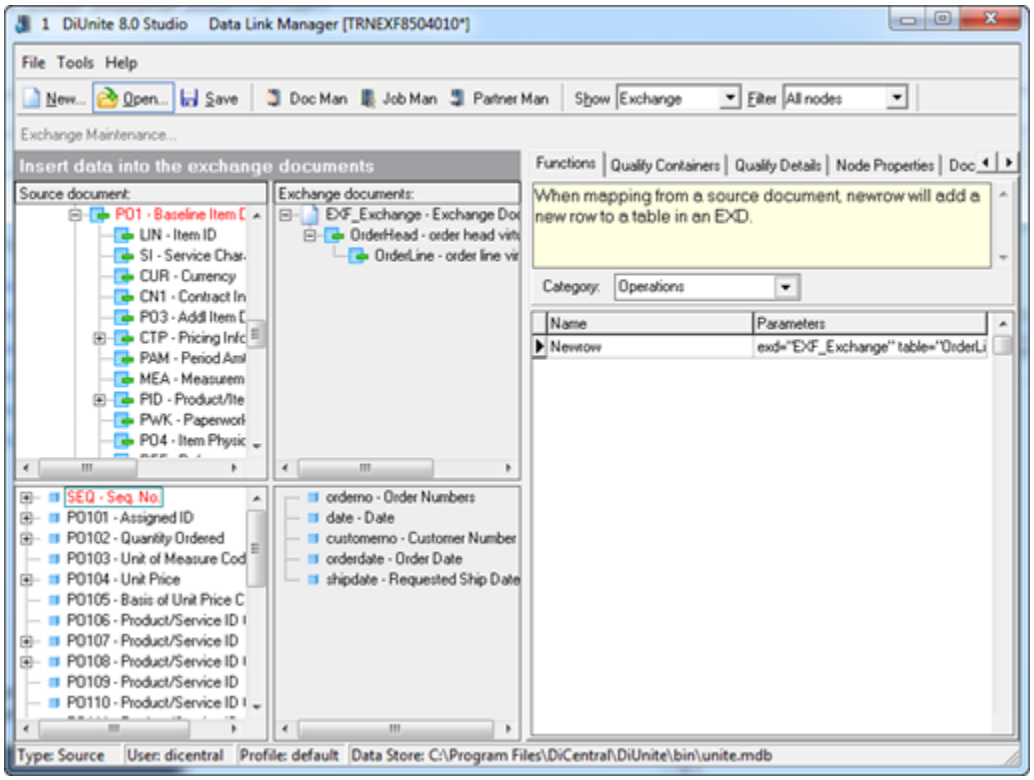
9. Select the **Name** field, enter **Requested Shipping Date,** and click **Add Qualifier**.

   A new DTM segment appears in the **Source document** window. Requested Shipping Date appears between the DTM detail node name and its description (Date/Time Ref).


1 DiUnite 8.0 Studio Data Link Manager [TRNEXF8504010*]
File Tools Help
New... Open... Save Doc Man Job Man Partner Man Show Exchange Filter All nodes
Exchange Maintenance...
Insert data into the exchange documents
Source document:
Exchange documents:
Functions Qualify Containers Qualify Details Node Properties Doc
Name: Requested Shipping Date
Add
Link: Default
Hierarchical Level:
Name Qualified Node Link
Purchase Order Date DTM
Available Qualifiers Description
SEQ Seq. No.
DTM01 Date/Time Qualifier
DTM02 Date
DTM03 Time
DTM04 Time Code
DTM05 Date Time Period Format Qualifier
DTM06 Date Time Period
Selected Qualifiers Value
DTM01 004
Type: Source User: dicentral Profile: default Data Store: C:\Program Files\DiCentral\DiUnite\bin\unite.mdb

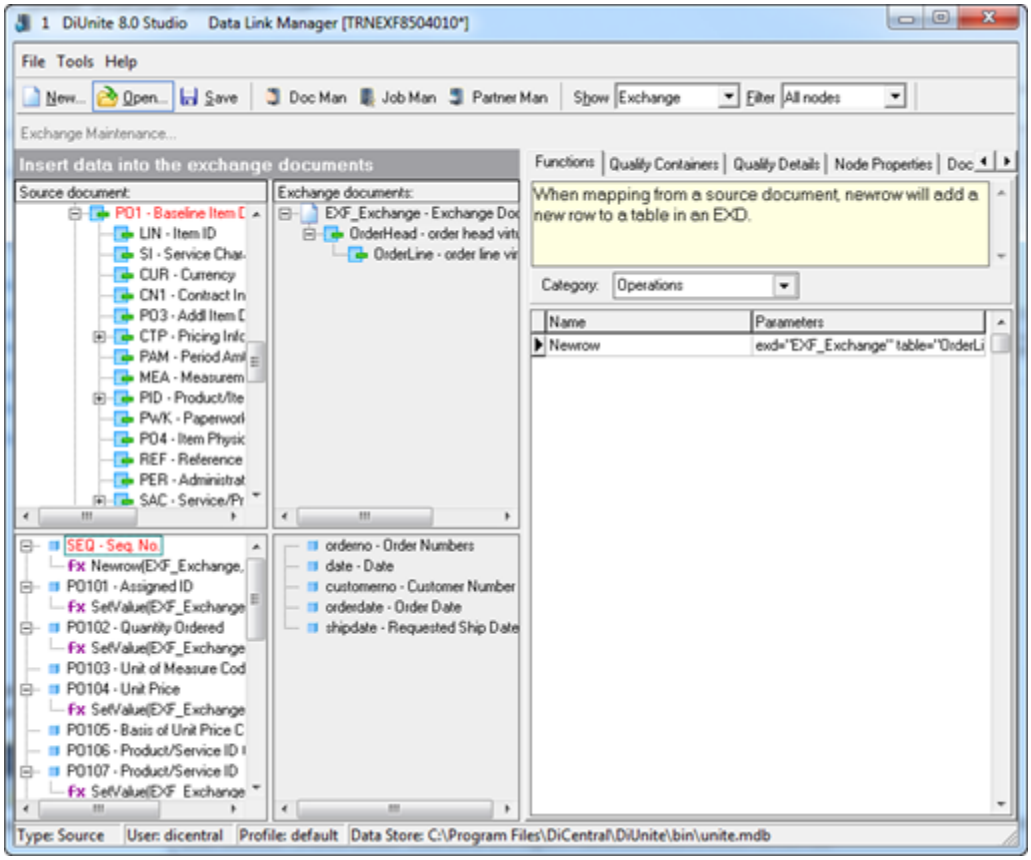
10. Click the **DTM [Requested Shipping Date]** container node.
11. Click the **DTM01** checkbox in the right side of the window to check it.

12. Select the **Value** field and enter “010” in the value field. “010”is the qualifier value for the Requested Shipping Date.


1 DiUnite 8.0 Studio Data Link Manager [TRNEXF8504010*]
File Tools Help
New... Open... Save Doc Man Job Man Partner Man Show Exchange Filter All nodes
Exchange Maintenance...
Insert data into the exchange documents
Source document:
Exchange documents:
Functions Qualify Containers Qualify Details Node Properties Doc
When mapping from a source document, newrow will add a new row to a table in an EXD.
Category: Operations
Name Parameters
Newrow exd="EXF_Exchange" table="OrderLi
SEQ - Seq No
PO101 - Assigned ID
PO102 - Quantity Ordered
PO103 - Unit of Measure Cod
PO104 - Unit Price
PO105 - Basis of Unit Price C
PO106 - Product/Service ID
PO107 - Product/Service ID
PO108 - Product/Service ID
PO109 - Product/Service ID
PO110 - Product/Service ID
orderno - Order Numbers
date - Date
customerno - Customer Number
orderdate - Order Date
shipdate - Requested Ship Date
Type: Source User: dicentral Profile: default Data Store: C:\Program Files\DiCentral\DiUnite\bin\unite.mdb

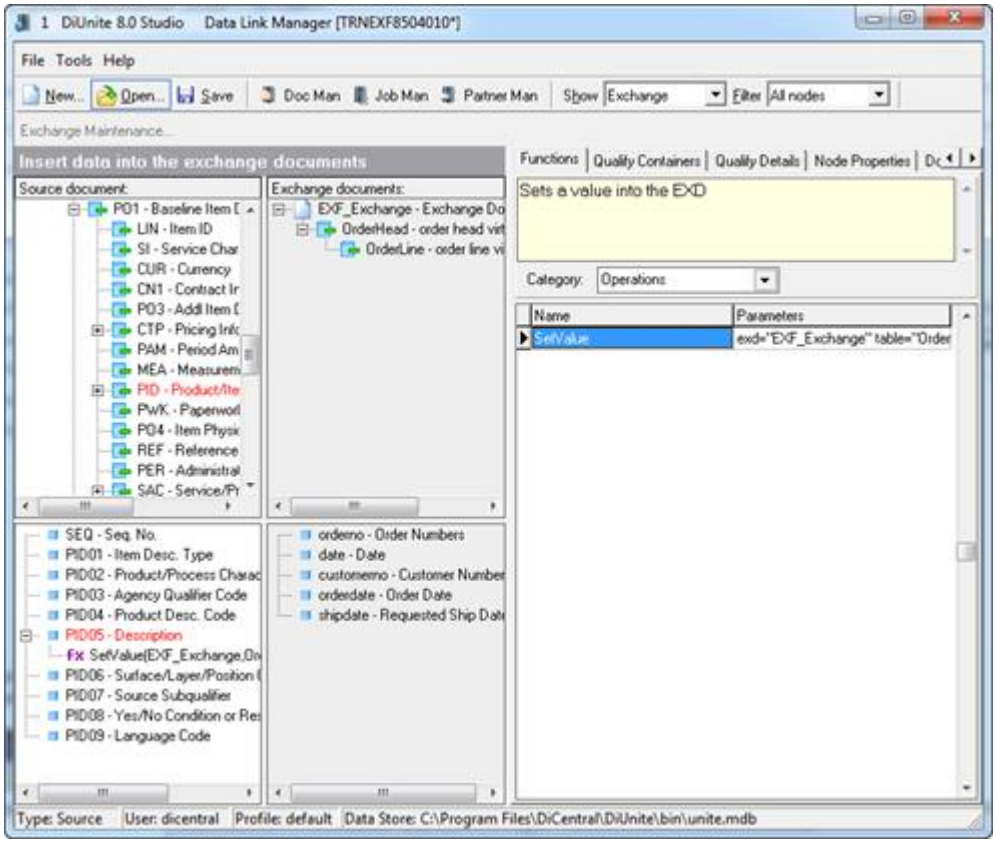
13. In the detail node pane of the source document working area, you can now map the Requested Shipping Date.
14. Click and drag the DTM02 detail node in the source document to the shipdate detail node in the exchange document.
15. Now, move down to the PO1 segment and set up the loop for the detail.
16. While highlighting the “SEQ – Seg No”, enter the Newrow function with the parameters EXD_Exchange and Orderline selected.
17. Drag the appropriate values from the 850 elements to the correlating elements in the exchange.

18. Select the Orderno from the BEG03 element in the 850 header to setvalue BEG03 to orderno.


1 DiUnite 8.0 Studio Data Link Manager [TRNEXF8504010*]
File Tools Help
New... Open... Save Doc Man Job Man Partner Man Show Exchange Filter All nodes
Exchange Maintenance...
Insert data into the exchange documents
Source document:
Exchange documents:
Functions Quality Containers Quality Details Node Properties Doc
When mapping from a source document, newrow will add a new row to a table in an EXD.
Category: Operations
Name Parameters
Newrow
SEQ - Seq. No.
Newrow(EXF_Exchange,
PO101 - Assigned ID
SetValue(EXF_Exchange
PO102 - Quantity Ordered
SetValue(EXF_Exchange
PO103 - Unit of Measure Cod
PO104 - Unit Price
SetValue(EXF_Exchange
PO105 - Basis of Unit Price C
PO106 - Product/Service ID
PO107 - Product/Service ID
SetValue(EXF Exchange
ordeno - Order Numbers
date - Date
customerno - Customer Number
orderdate - Order Date
shipdate - Requested Ship Date
Type: Source User: dicentral Profile: default Data Store: C:\Program Files\DiCentral\DiUnite\bin\unite.mdb

19. SetValue PO101 to lineno.
20. SetValue PO102 to quantity.
21. SetValue PO104 to price.

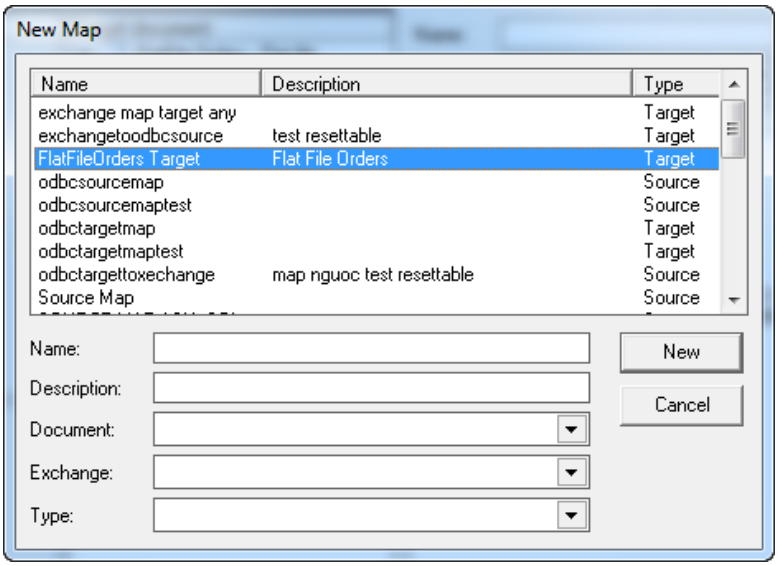
22. SetValue PO107 to itemno.


1 DiUnite 8.0 Studio Data Link Manager [TRNEXF8504010*]
File Tools Help
New... Open... Save Doc Man Job Man Partner Man Show Exchange Filter All nodes
Exchange Maintenance...
Insert data into the exchange documents
Source document:
Exchange documents:
Functions Quality Containers Quality Details Node Properties
Sets a value into the EXD
Category: Operations
Name Parameters
SetValue
Type: Source User: dicentral Profile: default Data Store: C:\Program Files\DiCentral\DiUnite\bin\unite.mdb

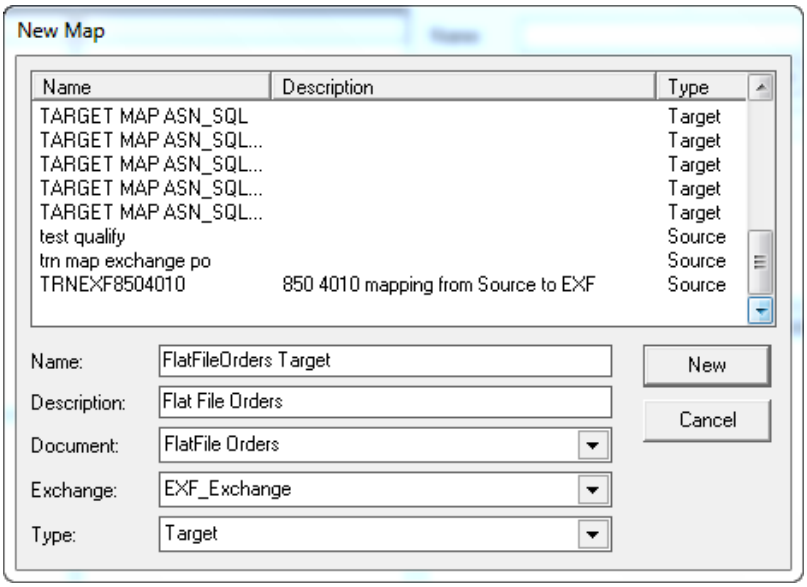
23. SetValue PID05 to itemdesc.
24. SetValue BEG05 to date.
25. Save Map.

4.3 Mapping from an Exchange Document to a Target Document

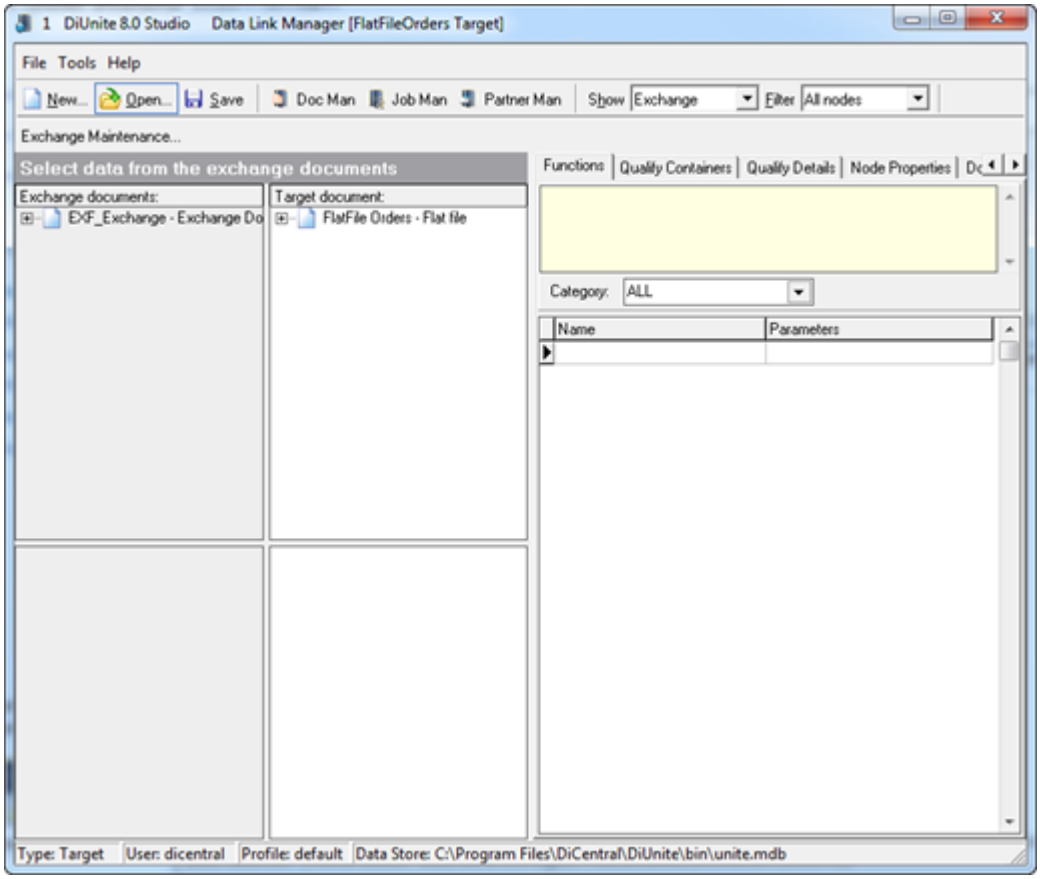
1. In the **Data Link Manager** window, click the **New** button. *The **New Map** window is displayed*.



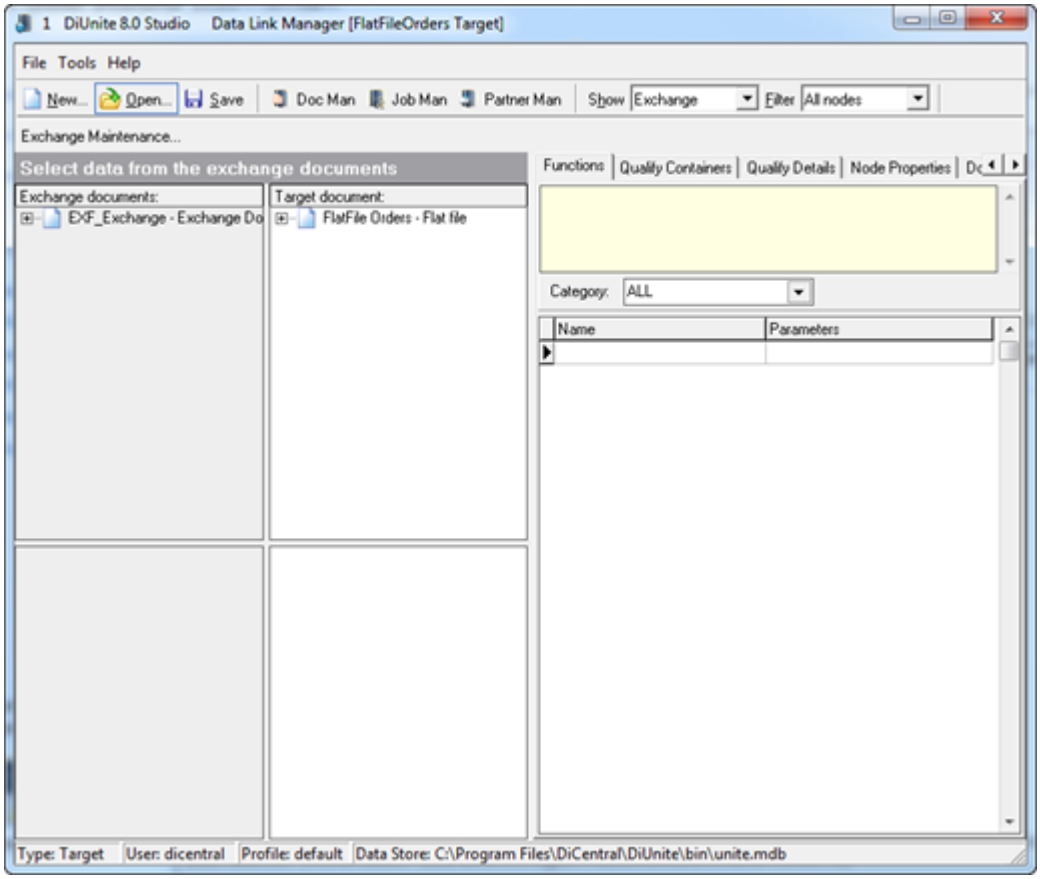
2. Select the **Name** field and enter “FlatFileOrders Target.”
3. Select the **Description** field and enter “Flat File Orders.”
4. Select **FlatFile ORDERS** from the **Document** dropdown list.
5. Select **EXD_Exchange** from the **Exchange** dropdown list. Notice that it is the same exchange document that you used to map from the source document.
6. Select **Target** from the **Type** dropdown list.



7. Click the **New** button. *The **Data Link Manager** screen is refreshed.*


1 DiUnite 8.0 Studio Data Link Manager [FlatFileOrders Target]
File Tools Help
New... Open... Save Doc Man Job Man Partner Man Show Exchange Filter All nodes
Exchange Maintenance...
Select data from the exchange documents
Exchange documents:
EXF_Exchange - Exchange Do
Target document:
FlatFile Orders - Flat file
Functions Qualify Containers Qualify Details Node Properties
Category: ALL
Name
Parameters
Type: Target
User: dicentral
Profile: default
Data Store: C:\Program Files\DiCentral\DiUnite\bin\unite.mdb

8. Click the **(+)** next to the source document node(s) to expand the source document and expose all the nodes.
9. Click the **(+)** next to the Exchange document node(s) to expand the exchange document and expose all the nodes.


1 DiUnite 8.0 Studio Data Link Manager [FlatFileOrders Target]
File Tools Help
New... Open... Save Doc Man Job Man Partner Man Show Exchange Filter All nodes
Exchange Maintenance...
Select data from the exchange documents
Exchange documents:
EXF_Exchange - Exchange Do
Target document:
FlatFile Orders - Flat file
Functions Qualify Containers Qualify Details Node Properties
Category: ALL
Name Parameters
Type: Target User: dicentral Profile: default Data Store: C:\Program Files\DiCentral\DiUnite\bin\unite.mdb

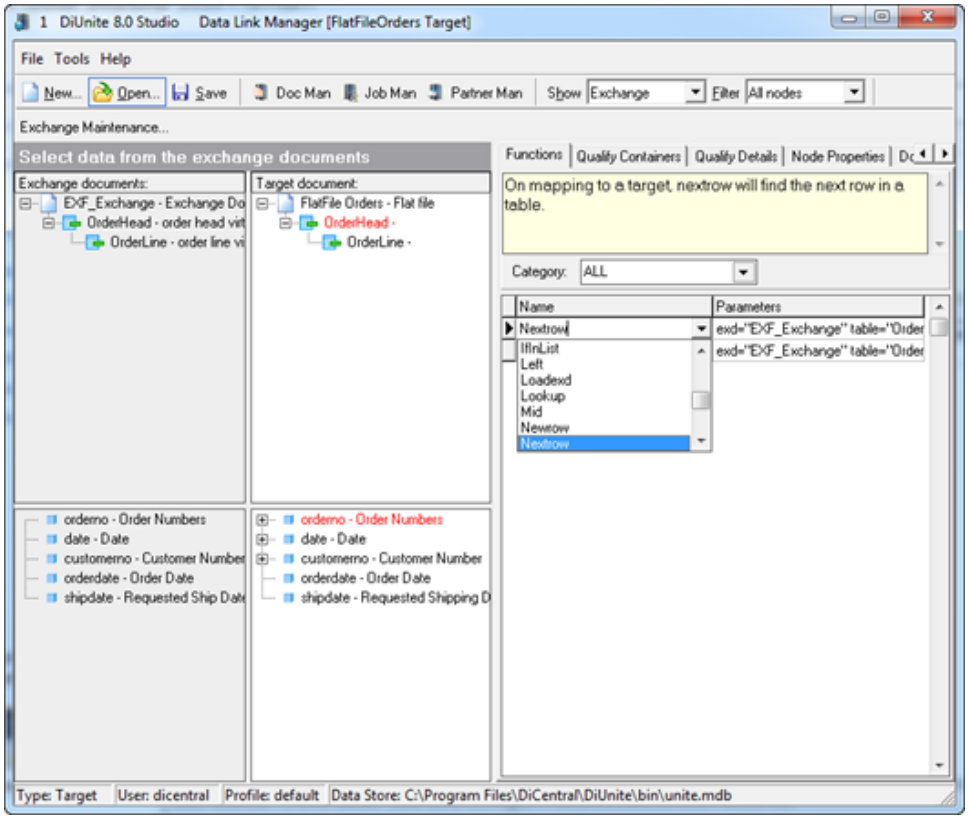
Remember, you will always map from Left to Right. When creating an Exchange to Target mapping, the **Exchange document** panes appear on the left and the mapping occurs to the **Target document** area on the right.

These are two simple structures that are very similar. For a Target document, you first insert a function that indicates where to start next record "Nextrow" for each table.

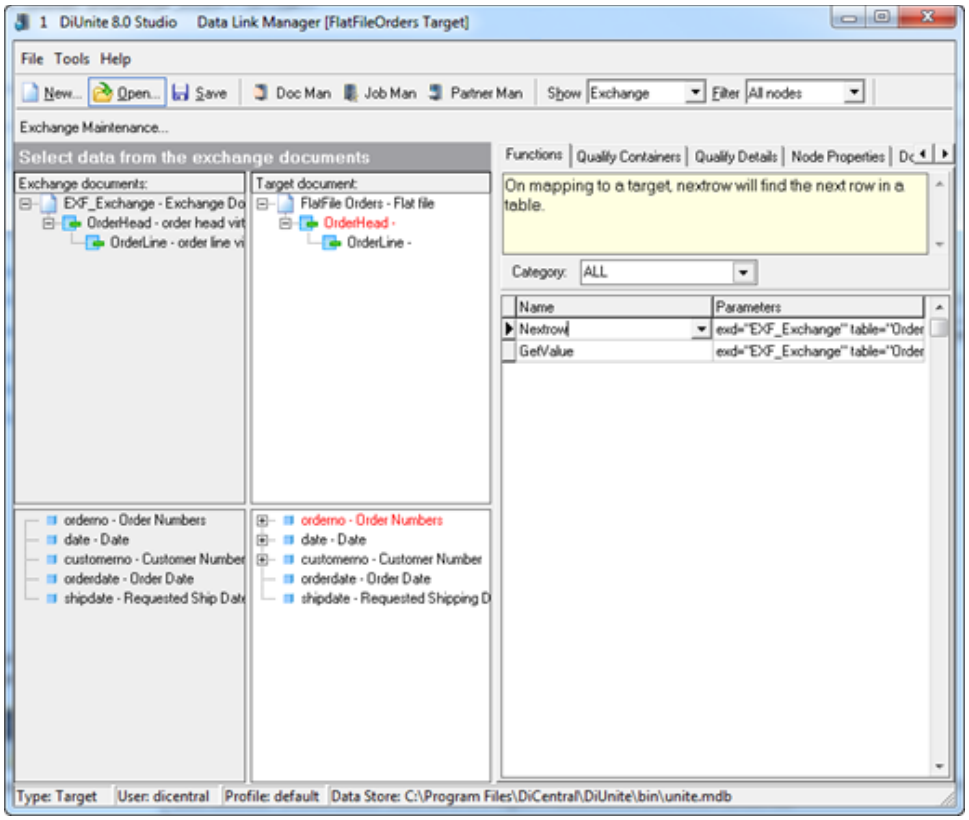
10. Click the **OrderHead** container node in the Exchange document to select it.
11. Click **OrderHeadRecord** in the Target document to select it.
12. Click the **orderno** detail node in the **Target document.**
13. Click the **Functions** tab.
14. Select **Target mapping** from the Category dropdown list.
15. Select the **Name** field and select **Nextrow** from the dropdown list. For every record you want to use from the Exchange Document, you should start a **Nextrow**.

    Again, the **Parameters** description displays in the **Functions** window. This is actually the opposite from the **Break Point** that was used in the Source map.

16. Select the **Parameters** field and **enter exd=”EXD_Exchange” table=”OrderHead”.**


1 DiUnite 8.0 Studio Data Link Manager [FlatFileOrders Target]
File Tools Help
New... Open... Save Doc Man Job Man Partner Man Show Exchange Filter All nodes
Exchange Maintenance...
Select data from the exchange documents
Exchange documents:
EXF_Exchange - Exchange Do
OrderHead - order head virt
OrderLine - order line vi
Target document:
FlatFile Orders - Flat file
OrderHead -
OrderLine -
Functions Quality Containers Quality Details Node Properties
On mapping to a target, nextrow will find the next row in a table.
Category: ALL
Name Parameters
Nextrow exd="EXF_Exchange" table="Order
IfInList
Left
Loadexd
Lookup
Mid
Newrow
Nextrow
orderno - Order Numbers
date - Date
customerno - Customer Number
orderdate - Order Date
shipdate - Requested Ship Dat
shipdate - Requested Shipping D
Type: Target User: dicentral Profile: default Data Store: C:\Program Files\DiCentral\DiUnite\bin\unite.mdb

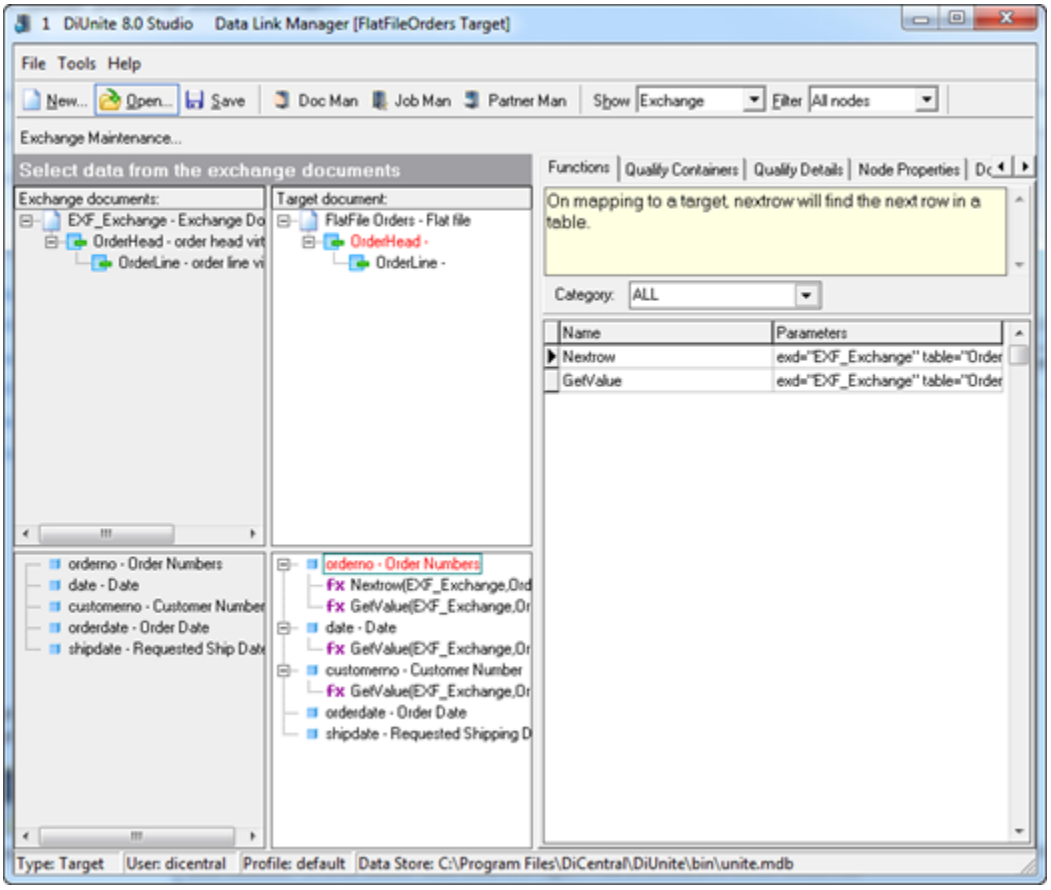
17. Click the **orderno** detail node in the Exchange document, and drag-and-drop it over the **orderno** detail node in the Target document.


1 DiUnite 8.0 Studio Data Link Manager [FlatFileOrders Target]
File Tools Help
New... Open... Save Doc Man Job Man Partner Man Show Exchange Filter All nodes
Exchange Maintenance...
Select data from the exchange documents
Exchange documents:
EXF_Exchange - Exchange Do
OrderHead - order head virt
OrderLine - order line vi
Target document:
FlatFile Orders - Flat file
OrderHead -
OrderLine -
Functions Quality Containers Quality Details Node Properties
On mapping to a target, nextrow will find the next row in a table.
Category: ALL
Name Parameters
Nextrow exd="EXF_Exchange" table="Order
GetValue exd="EXF_Exchange" table="Order
orderno - Order Numbers
date - Date
customerno - Customer Number
orderdate - Order Date
shipdate - Requested Ship Dat
shipdate - Requested Shipping D
Type: Target User: dicentral Profile: default Data Store: C:\Program Files\DiCentral\DiUnite\bin\unite.mdb

18. Select the **OrderLine** container node in the Target document.
19. Click the **orderno** detail node in **the Exchange document**.
20. Select the **Name** field of the **Functions** work area and select **NextRow** from the dropdown list. For every record you want to use in the exchange document, you should start a NextRow.

    Again, the **Parameters** description displays in the **Functions** window. This is actually the opposite from the Break Point that was used in the Source map.

21. Select the **Parameters** field and enter **exd="EXD_ORDERS" table="OrderLine"**.
22. Select each field from the Exchange document and drop them on the appropriate field in the Target document working area.
    *The **GetValue** function appears in the Functions working area for each of the nodes.*



23. Click the **Save** button to save the map. Select **Exit** from the **File Menu**.

5. Creating a Job

A job is a list of tasks that occur in order for a document conversion to take place once the mapping has been completed. In Job Manager, this means combining the mappings and converting data. Once you have created a job, you then generate an .xml file, which is what the DiUnite Runtime needs to perform the actual conversion.

5.1 Starting Job Manager

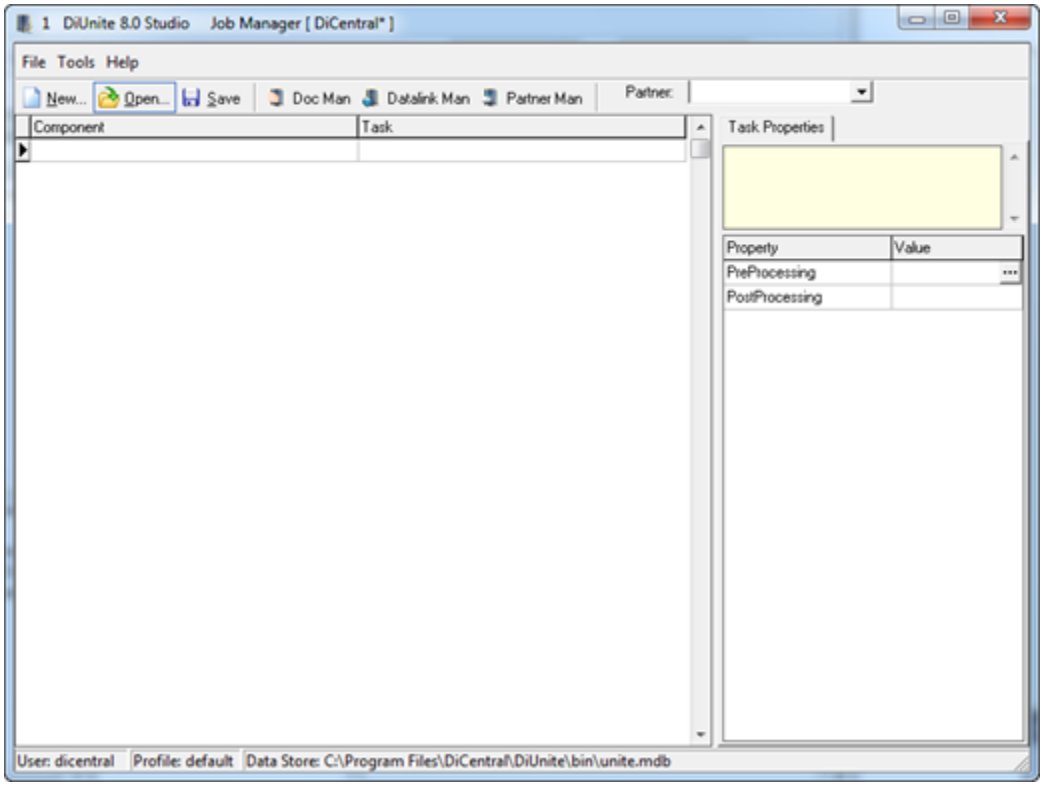
To start Job Manager:

1. Click the Windows **Start** button. Next, point to **All Programs** > **DiCentral** > **DiUnite** and then click **Job Manager**. *The **Login** window is displayed*.

DiUnite 8.0 Studio Login
DiUnite 8.0 Studio Job Manager
User name:
dicentral
Password:
OK
Cancel
Options >>
©Copyright 2007 - 2011 DiCentral Corporation.
All rights reserved. This program is protected by US and international copyright laws as described in Help About.

2. Select the **User Name Field** and enter your user name.
3. Select the **Password** field and enter your password.

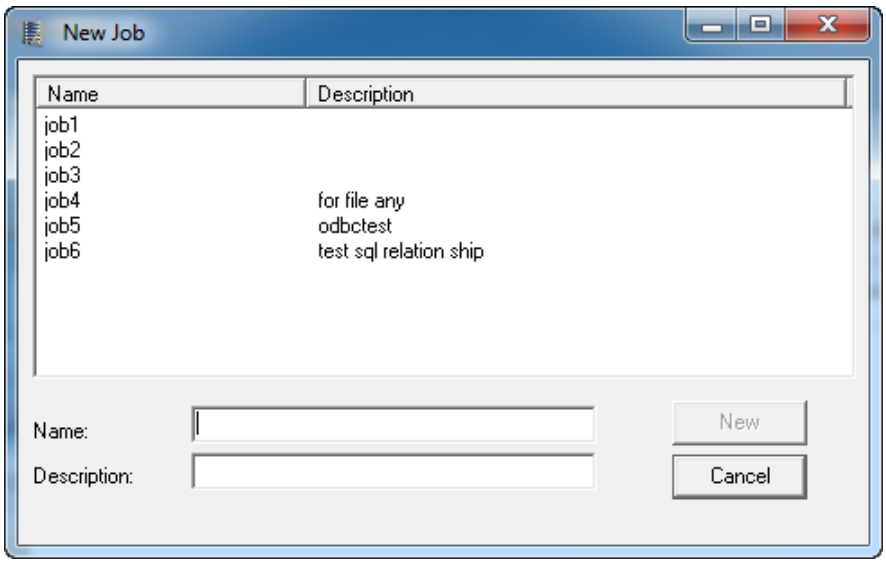
4. Click the **OK** button. *The **Job Manager** window is displayed*.



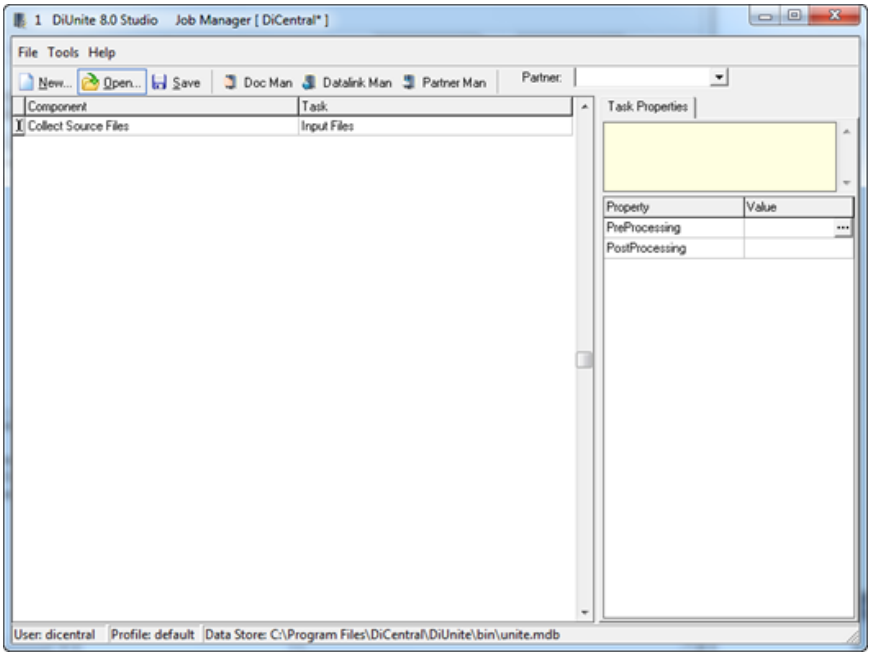
5.2 Creating a New Job

Creating a new job is a simple process that involves selecting which document you want to copy from and which document you want to copy to.

1. Click **New** in the Job Manager's Main window. *The New Job window is displayed*.



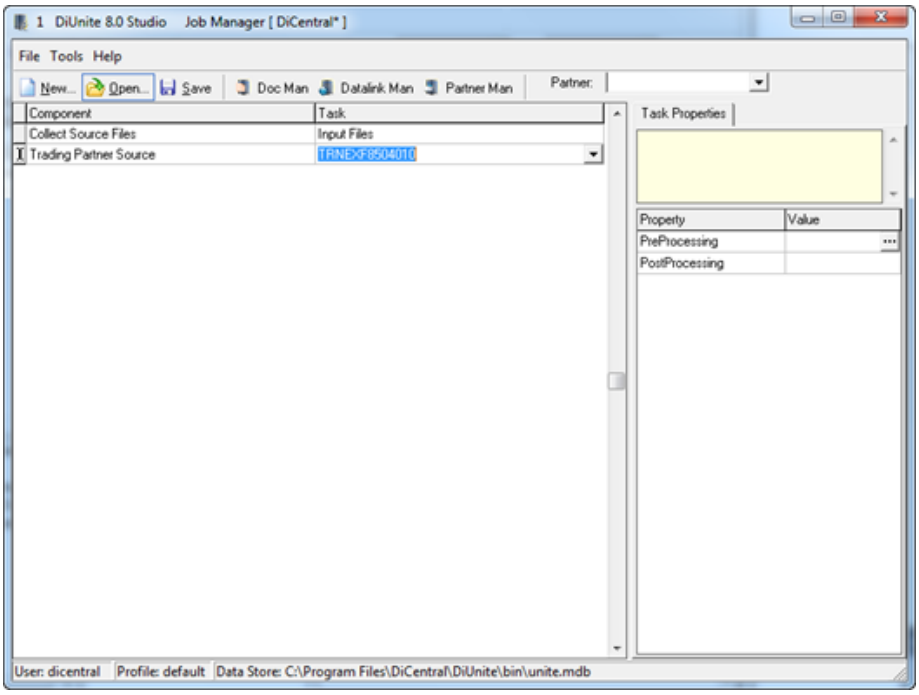
2. Select the **Name** field and enter "DiCentral".
3. Select the **Description** field and enter "Tutorial Job".
4. Click the **New** button to create the job. *The name of the job appears in the Title Bar*.


1 DiUnite 8.0 Studio Job Manager [DiCentral*]
File Tools Help
New... Open... Save Doc Man Datalink Man Partner Man Partner:
Component Task
Collect Source Files Input Files
Task Properties
Property Value
PreProcessing
PostProcessing
User: dicentral Profile: default Data Store: C:\Program Files\DiCentral\DiUnite\bin\unite.mdb

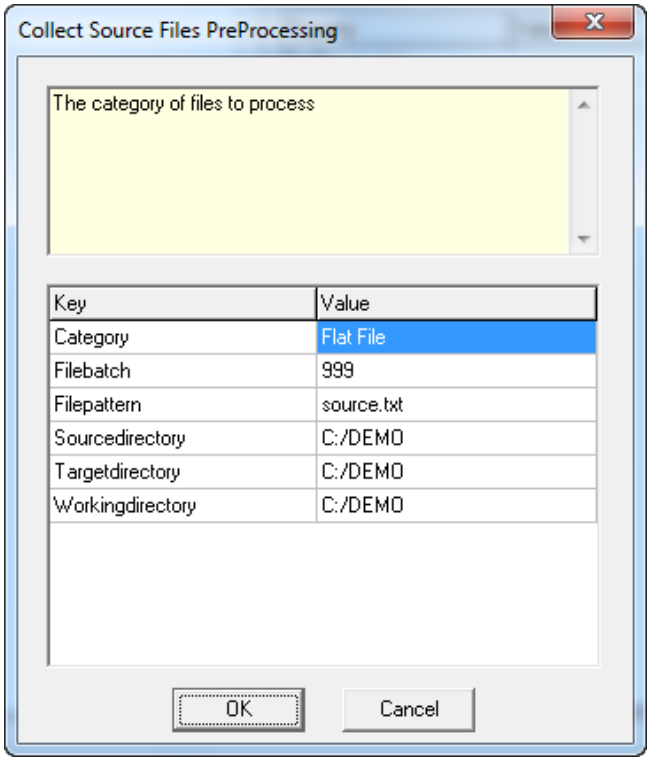
Now, select all the Components you want to include in the current job and place them in the order you want to process during the job. This process collects all the TRN files you want to process from DiTranslator.

5.3 Setting up the Job

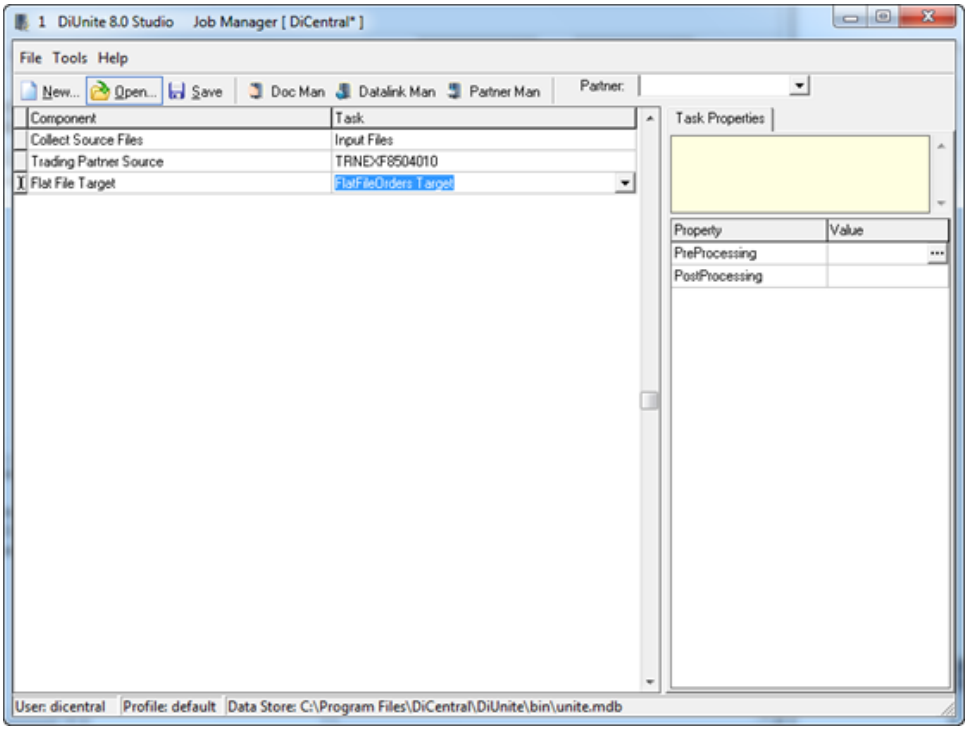
1. Set up **Collect Source File Component**:
   a. Select the **Component** column. *An arrow appears in the selected box*.
   b. Click the arrow and select **Collect Source Files** from the dropdown list.
   c. Select the **Task** column and enter “Input File”.


1 DiUnite 8.0 Studio Job Manager [DiCentral*]
File Tools Help
New... Open... Save Doc Man Datalink Man Partner Man Partner:
Component Task
Collect Source Files Input Files
Trading Partner Source
Task Properties
Property Value
PreProcessing
PostProcessing
User: dicentral Profile: default Data Store: C:\Program Files\DiCentral\DiUnite\bin\unite.mdb

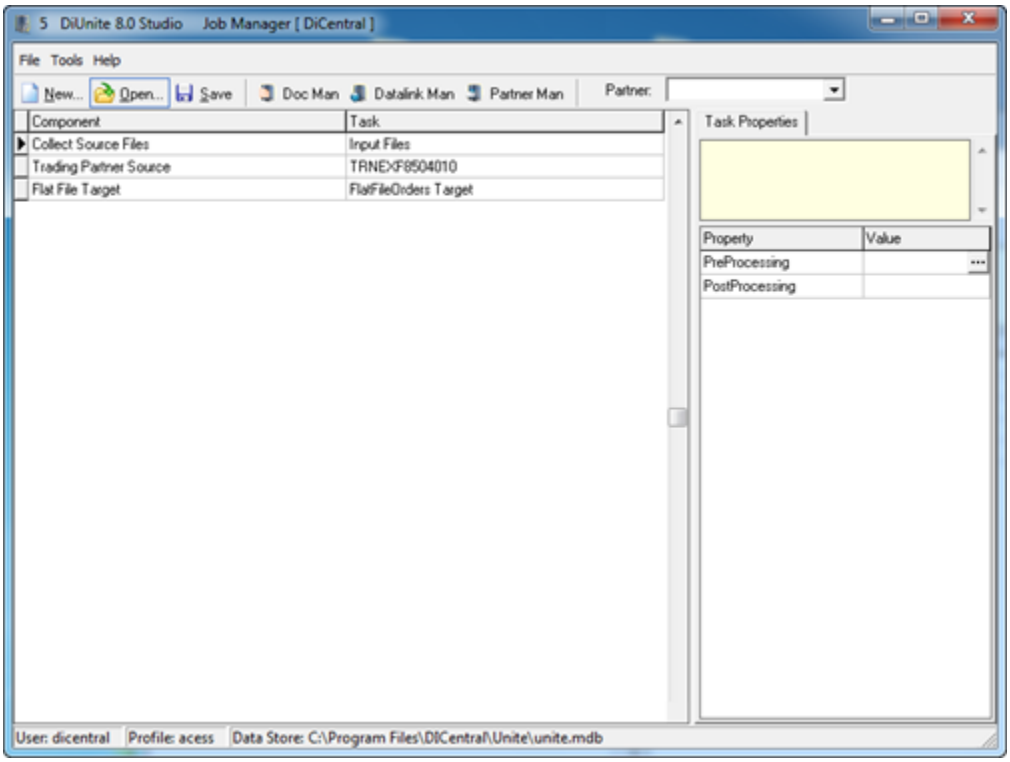
d. Select **PreProcessing** to establish the directory and file name for input data.

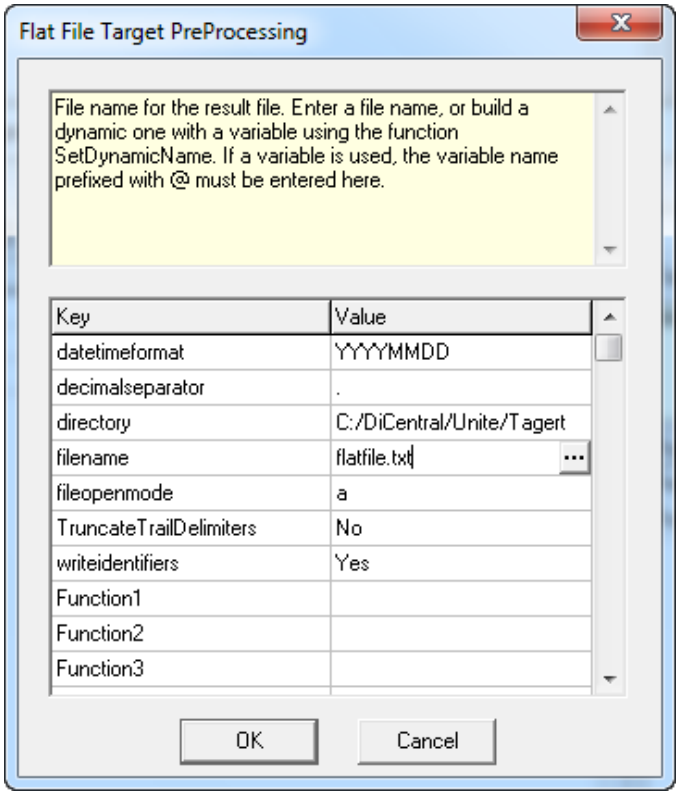


2. Set up **DiTranslator Source Component**:
   a. Press the down arrow key. *A row is added.*
   b. Select the **Component** column and select **DiTranslator Source Component**.
   c. Select the corresponding box in **Task** column. *An arrow appears*.
   d. Click the arrow and select **TRNEXD4010** from the dropdown list. This is the source mapping you just created.

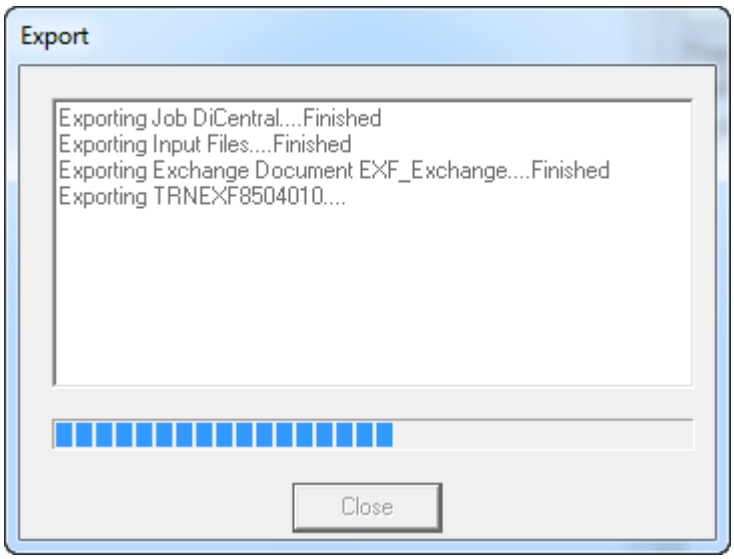


3. Set up **Flat File Target Component**:
   a. Press the down arrow key. *A row is added*.
   b. Select the **Component** column and select **Flat File Target Component**.
   c. Select the corresponding box in the **Task** column and select **FlatFileOrders Target** from the dropdown list.
   d. Select **PreProcessing** to establish output directory and filename.





4. Click the **Save** button to save the job.
5. Select **Export > All Definitions to Metafile** from the File menu. *The Export window appears and displays the progress of the job*.



6. When the job finishes, click the **Close** button.

You can now use **jobrun** to run the conversion. The export generates .xml files (map files) located in "C:\DiCentral\DiUnite\jobs" (or where you defined the location in the DiUnite.ini file).

These .xml files are what the DiUnite Runtime needs to run the conversion.

Note:
Before simulating an actual run, you will need to have test input data.

Here is an example command from a batch file:

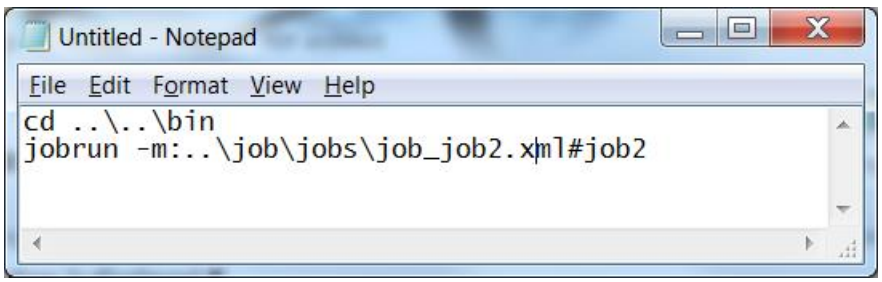
```
cd "\DiCentral\DiUnite\Bin"

jobrun -m:..\jobs\xyz\job_xyz.xml#xyz
```

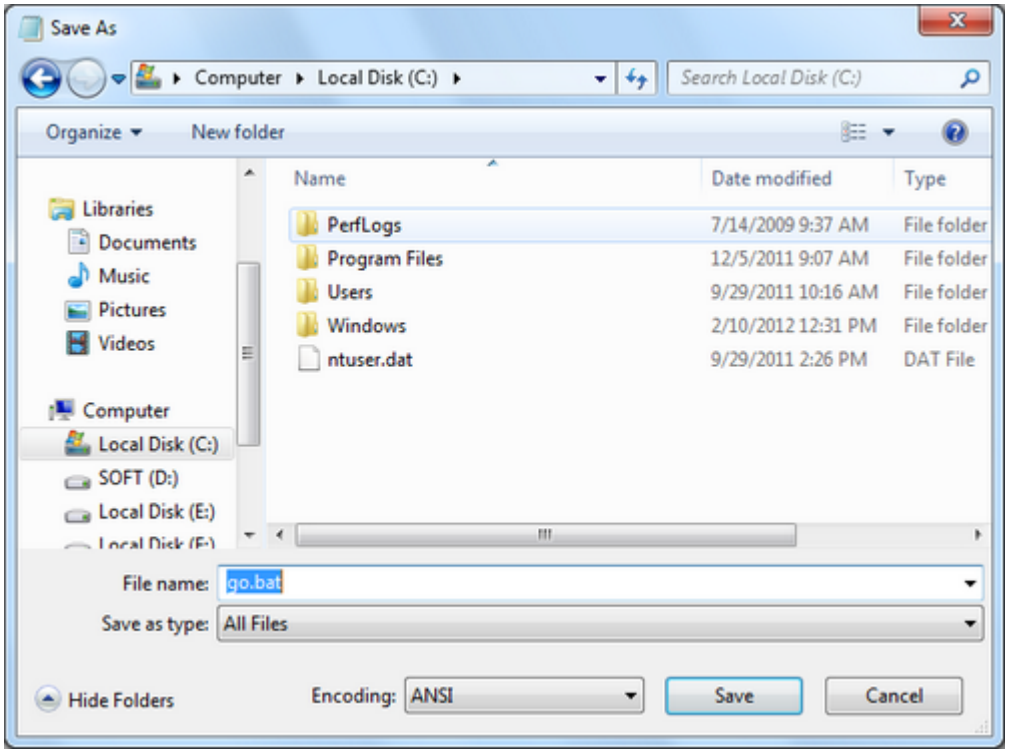
We need to create a batch file that executes a job.

To create a batch file

1. Click the Windows **Start** button. Next, point to **All Programs** > **Accessories** and then click **Notepad**. *The Notepad window is displayed.*
2. Type the following text:



3. Select **Save As** from the File menu.
4. In the **Save As** dialog box,
   a. Type the file name with an extension of .bat in the **File name** box.
   b. Select "**All Files**" from the **Save as type** dropdown list.



   c. Click the **Save** button.

The "go.bat" file is explained below:

1. First line: Change the location of the DiUnite Code.
2. Second line: Execute **jobrun** started by the dash (-), followed by an "m" and location where a job was exported. In most cases, the jobs are exported out of a subdirectory of the DiUnite installation. The job name, particular XML file, #sign, and the job are followed.

Appendix A: The Functions

1. Data Manipulation

1.1 Abs []

Sets the current value to the positive value of itself. If the current value is a string or a date, it will not do anything.

1.2 AddDate [name | value | datepart]

Adds a datetime variable.

- name (@currentvalue): The name of date variable to add to.
- value: The value to be added.
- datepart (year|month|day|hour|minute|second): The time part to be added.

1.3 Convert [datatype | format]

Converts the current value into any format.

- datatype (text|integer|money|datetime|float): The datatype to convert the value into.
- format (optional): The format mask that will be used to format the current value.

1.4 DefaultValue [value]

Sets a default value in a function list. If the previous function (or the message) returns a value, this value will be used. Otherwise, the default value will be used.

- value: The value that should be set into the variable.

1.5 DelimitedMid [value | start | end | fieldseparator | fieldno | key | keyseparator | outputvariable]

Extracts a substring out of the current string.

- value (optional): The value that should be searched. If not specified, the current value will be used.
- start (optional): The string that identifies where the substring should start from.
- end (optional): The string that identifies where the substring should end.
- fieldseparator (optional): The parameter that specifies the separator between fields, if retrieving a field.
- fieldno (optional): The parameter that specifies the field number, if retrieving a field.
- key (optional): The parameter that specifies the key being searched, if retrieving a field and the field has a key.
- keyseparator (optional): The parameter that specifies the key separator, if retrieving a field and the field has a key.

- outputvariable (optional): If specified, the result will be placed into this variable. The current value will then get the value 1 if a value was found, and 0 if not. If an output variable is not specified, the result will be sent to the current value.

1.6 Find [string | position | startposition | replace | replacestring]

Returns the position of the substring in a main string.

- string: The string to search for.
- position (optional): The variable where the resulting position should be entered. Should start with @.
- startposition (optional): The location where the search should start. 1 is the first position.
- replace (none|first|all|recursive)(optional): The type of replace to be used. Default is 'none'.
- replacestring (optional): The string to replace with.

1.7 FixedValue [value]

Sets a new current value even if a current value already exists. Works similar to DefaultValue.

- value: The value to be used.

1.8 Format [format]

Formats the current value into a text string.

- format: The format mask that will be used to format the current value.

1.9 InsertDelimiter [sections | value | start | end | fieldseparator | key | keyseparator]

Inserts or replaces a substring into the current string.

- sections (optional): The string that should be used. If not specified, the current value will be used.
- value (optional): The value that should be inserted.
- start (optional): The string that identifies where the substring should start from. If the start cannot be found, then this string will be inserted.
- end (optional): The string that identifies where the substring should end.
- fieldseparator (optional): The parameter that specifies the separator between fields, if inserting a field.
- key (optional): The parameter that specifies the key being inserting, if inserting a field and the field has a key.
- keyseparator (optional): The parameter that specifies the key separator, if inserting a field and the field has a key.

1.10 Left [length | value]

Extracts a substring from the beginning of the current string.

- length: The length of the substring.
- value (optional): The left part of the string value. Default is '@currentvalue'.

1.11 Length [value]

Returns the length of a text.

- value: The value to get the length of.

1.12 LowerCase[variable]

Converts the string variable into lowercase.

- variable (optional): The string variable.

1.13 Mid [position | length | value]

Extracts a substring from the current string.

- position: The start position of the substring within the main string. The position must be smaller than the length.
- length (optional): The length of the substring. If negative or not specified, it will get the rest of the string.
- value (optional): The string value to extract a string from. Default is '@currentvalue'.

1.14 Now []

Returns a date-time with current date and time.

1.15 Padding [length | side | symbol | variable]

Adds padding to the left or right of a string to make it a specific length.

- length: The length that the result string should have.
- side (left|right): The side that will be padded. Default is 'left'.
- symbol (optional): The symbol that will be filled. Default is 'space'.
- variable (optional): The variable that should be padded. Default is '@currentvalue'.

1.16 Right [length | value]

Extracts a substring from the end of the current string.

- length: The length of the substring.
- value (optional): The right part of the string value. Default is '@currentvalue'.

1.17 Trim [value | side]

Trims away space from the beginning and end of the current string value.

- value (optional): The string value to be trimmed. Default is '@currentvalue'.
- side (both|left|right)(optional): The side of the string that should be trimmed.

1.18 UpperCase[variable]

Converts the string variable into uppercase.

- variable (optional): The string variable.

2. Source Mapping

2.1 NewRow [exd | table]

Adds a new row to a table in an EXD. NewRow is used in source mapping.

- exd: The name of the EXD.
- table: The name of table.

2.2 SetValue [exd | table | field | value | template]

Sets a value into the EXD.

- exd: The name of the EXD.
- table: The name of the table.
- field: The name of the field.
- value (optional): The value to be set into the EXD. If the value is not set, it will use the returnvalue from the last function.
- template (true|false)(optional): The template should be set when mapping to a field before the NewRow function. The value that is set will then be put into the template and added for each NewRow function thereafter. Default is 'false'.

2.3 RemoveDuplicateRow [exd | table | field1 | field2 | field3 | field4 | field5 | sumField1 | sumField2 | sumField3 | sumField4 | sumField5]

Removes the current row if a row with the same key values already exists. RemoveDuplicateRow is used in source mapping.

- exd: The name of the EXD.
- table: The name of the table.
- field1: The name of the first key field.
- field2 (optional): The name of the second key field.
- field3 (optional): The name of the third key field.

- field4 (optional): The name of the fourth key field.
- field5 (optional): The name of the fifth key field.
- sumField1 (optional): The name of the first field that needs to be summed.
- sumField2 (optional): The name of the second field that needs to be summed.
- sumField3 (optional): The name of the third field that needs to be summed.
- sumField4 (optional): The name of the fourth field that needs to be summed.
- sumField5 (optional): The name of the fifth field that needs to be summed.

2.4 RemoveEmptyRow [exd | table | field1 | field2 | field3 | field4 | field5]

Removes the current row if the key fields are empty. RemoveEmptyRow is used in source mapping.

- exd: The name of the EXD.
- table: The name of the table.
- field1: The name of the first key field.
- field2 (optional): The name of the second key field.
- field3 (optional): The name of the third key field.
- field4 (optional): The name of the fourth key field.
- field5 (optional): The name of the fifth key field.

3. Target Mapping

3.1 NextRow [exd | table]

Find the next row in a table. The positioning of this function will indicate a looping point of the message. NextRow is used in target mapping.

- exd: The name of the EXD.
- table: The name of the table.

3.2 GetRow [exd | table | field | value]

Searches through a given EIF table for a specific field value in a row. Sets @currentvalue to 1 if found, and 0 if not found.

- exd: The name of the EXD.
- table: The name of the table.
- field: The name of the field.
- value: The key value search for.

3.3 FindNextRow [exd | table | field | value]

Works similar to NextRow, but will additionally set a 'find' criteria. Finds the next row in a table that matches the 'find' criteria or description. The difference between GetRow and FindNextRow is that GetRow is used to find the first row that matches the criteria, while FindNextRow will loop through all the rows that match. FindNextRow is used in target mapping.

- exd: The name of the EXD.
- table: The name of the table.
- field: The name of the field.
- value: The key value you want to search.

3.4 GetValue [exd | table | field | aggregate]

Gets a value from the EXD.

- exd: The name of the EXD.
- table: The name of the table field.
- field: The name of the field.
- aggregate (none|sum|min|max|count|average|first|last)(optional): Retrieves data from a table that is not yet selected with NextRow, FindNextRow or GetRow. Since there may be several rows, an aggregate function needs to be chosen.

3.5 ContinueMessage []

For Flat File Target Only! Skips one lap in the loop/group and continues with the next one.

4. Conversion Functions

4.1 AddConversion [fromvalue | tovalue | position]

Adds a static conversion to the function processing. This conversion will be run for all strings sent to and from function processing. This functionality will typically be used when a character in a source document should be converted into something else throughout the entire source document. When several conversions are added, they will all be used at the same time, either before or after function processing. Remember to run the Clearconversion function when conversion is not needed anymore.

- fromvalue: The string value that should be converted from.
- tovalue: The string value that should be converted to.
- position: The position of the string where the search should start from.

4.2 AddStringConversion [list | fromvalue | tovalue]

Adds a static conversion to the function processing. Full string conversion. This conversion can be organized into several lists and will be called every time it is used with the StringConvert function. One conversion list may be a unit description list, where you may want to convert all UNITS into PCE, LITRE to LTR... All these conversions can be added to the same list. When a conversion of this type is needed, only this list will be called. Remember to run the Clearconversion function when conversion is not needed anymore.

- list: The conversion list that the conversion should be added to.
- fromvalue: The string value that it should be converted from.
- tovalue: The string value that it should be converted to.

4.3 StringConvert [list | default | casesensitive]

Converts the current string value to another string according to an internal conversion list.

- list: The name of the conversion list to use.
- default: The value that will be used if no other match is found.
- casesensitive (true|false): True if the test should be case sensitive.

4.4 Clearconversion []

Removes all conversions that have been set with the AddConversion function.

4.5 Lookup [filename | section | casesensitive | load | key | default]

Looks up the current value in an x-ref list and finds the corresponding value.

- filename: The name of the file where the x-ref list is and can set the variable.
- section: The section, within the file, where the list is and can set the variable.
- casesensitive (true|false): Should the comparison be case sensitive?
- load (true|false): Should the x-ref list be loaded into memory?
- key (optional): A hardcoded value to search for. Default is ‘current value’.
- default (optional): The value that will be set if the current value was not found in the list.

4.6 LookupAdv [filename | recordseparator | fieldseparator | keyfieldno | valuefieldno | casesensitive | load | key | default]

Looks up the current value in an x-ref list and finds the corresponding value.

- filename: The name of the file where the x-ref list is.
- recordseparator (optional): The symbol that separates records.
- fieldseparator (optional): The symbol that separates fields.
- keyfieldno (optional): The field that holds the key value.
- valuefieldno (optional): The field that holds the return value.

- casesensitive: Should the comparison be case sensitive?
- load: Should the x-ref list be loaded into memory?
- key (optional): The key where the list is within the section. Default is 'current value'.
- default (optional): The value that will be set if current value was not found in the list.

5. Exchange

5.1 SetFilter [exd | table | field | value | comparetype]

Sets a filter on an EXD.

- exd: The name of the EXD.
- table: The name of the table.
- field: The name of the field.
- value: The value to be compared with the field value.
- comparetype (= | != | >= | <=): The comparison operator.

5.2 ClearFilter []

Clears all filters in the EXD.

5.3 ClearFilterAll []

Clears all filters in all EXDs and removes the automatic filter that filters away messages with errors.

5.4 GroupTable [exd | table | field1 | field2 | field3 | field4 | field5 | sumField1 | sumField2 | sumField3 | sumField4 | sumField5]

Sums similar rows into one row.

- exd: The name of the EXD.
- table: The name of the table.
- field1: The name of the first key field.
- field2 (optional): The name of the second key field.
- field3 (optional): The name of the third key field.
- field4 (optional): The name of the fourth key field.
- field5 (optional): The name of the fifth key field.
- sumField1 (optional): The name of the first field that needs to be summed.
- sumField2 (optional): The name of the second field that needs to be summed.
- sumField3 (optional): The name of the third field that needs to be summed.
- sumField4 (optional): The name of the fourth field that needs to be summed.
- sumField5 (optional): The name of the fifth field that needs to be summed.

5.5 ResetTable [exd | table]

Resets the counter for a given table and all the children. Used if you want to loop the same dataset more than once.

- exd: The name of the EXD.
- table: The name of the table.

6. Flow Control

6.1 AndIf [lvalue | rvalue | operator | datatype]

Does a comparison and lets processing continue to the next function if the comparison is true. If the comparison is false, processing will look for endif, else or elseif.

- lvalue: The value on the left side of the comparison.
  Legal values:
  1. @currentvalue indicates that the value returned from the previous function.
  2. @othervalue indicates that the value should be retrieved from the variable 'othervalue'.
  3. Values not starting with @ will be used as is.
- rvalue: The value on the right side of the comparison.
  Legal values:
  1. @currentvalue indicates that the value returned from the previous function.
  2. @othervalue indicates that the value should be retrieved from the variable 'othervalue'.
  3. Values not starting with @ will be used as is.
- operator (= | != | >= | <=): The comparison operator.
- datatype (text|integer|money|float|datetime)(optional): The datatype that the values should be when they are compared.

6.2 Break []

Exits the current While ... EndWhile or Do ... LoopWhile section before its normal ending.

6.3 ContinueLoop []

Skips the rest of the functions in a WhileLoop and starts from the top again.

6.4 Do []

Marks the start of a Do…LoopWhile-section. Works similar to While…EndWhile, but will always allow the code to run once, since the logical test is in the end.

6.5 Else []

Marks the end of the ‘true’-part of an if-section and starts the ‘false’-part of the if-section.

6.6 ElseIf [lvalue | rvalue | operator | datatype]

Does a comparison and lets processing continue to the next function if the comparison is true. If the comparison is false, processing will look for endif, else, or elseif.

- lvalue: The value on the left side of the comparison.
  Legal values:
  1. @currentvalue indicates that the value returned from the previous function.
  2. @othervalue indicates that the value should be retrieved from the variable 'othervalue'.
  3. Values not starting with @ will be used as is.
- rvalue: The value on the right side of the comparison.
  Legal values:
  1. @currentvalue indicates that the value returned from the previous function.
  2. @othervalue indicates that the value should be retrieved from the variable 'othervalue'.
  3. Values not starting with @ will be used as is.
- operator (= | != | >= | <=): The comparison operator.
- datatype (text|integer|money|float|datetime)(optional): The datatype that the values should be when they are compared.

6.7 EndIf []

Marks the end of the if-section.

6.8 EndWhile []

Marks the end of a while-section.

6.9 If [lvalue | rvalue | operator | datatype]

Does a comparison and lets processing continue to the next function if the comparison is true. If the comparison is false, processing will look for endif, else or elseif.

- lvalue: The value on the left side of the comparison.
  Legal values:
  1. @currentvalue indicates that the value returned from the previous function.
  2. @othervalue indicates that the value should be retrieved from the variable 'othervalue'.
  3. Values not starting with @ will be used as is.
- rvalue: The value on the right side of the comparison.
  Legal values:
  1. @currentvalue indicates that the value returned from the previous function.
  2. @othervalue indicates that the value should be retrieved from the variable 'othervalue'.
  3. Values not starting with @ will be used as is.
- operator (= | != | >= | <=): The comparison operator.
- datatype (text|integer|money|float|datetime)(optional): The datatype that the values should be when they are compared.

6.10 IfInList [filename | section| casesensitive | load]

Looks for the current value in an x-ref list. If the value is found, processing will continue to the next function. If the value is not found, processing will look for endif, else, or elseif.

- filename: The name of the file where the x-ref list is
- section: The section, within the file, where the list is.
- casesensitive: Should the comparison be case sensitive?
- load: Should the x-ref list be loaded into memory?

6.11 Iif [lvalue | rvalue | operator | datatype | truepart | falsepart]

Does a comparison and returns the true part if the comparison is a success. If the comparison is false, it will return the false part.

- lvalue: The value on the left side of the comparison.
  Legal values:
  1. @currentvalue indicates that the value returned from the previous function.
  2. @othervalue indicates that the value should be retrieved from the variable 'othervalue'.
  3. Values not starting with @ will be used as is.
- rvalue: The value on the right side of the comparison.
  Legal values:
  1. @currentvalue indicates that the value returned from the previous function.
  2. @othervalue indicates that the value should be retrieved from the variable 'othervalue'.
  3. Values not starting with @ will be used as is.
- operator (= | != | >= | <=): The comparison operator.
- datatype (text|integer|money|float|datetime)(optional): The datatype that the values should be when they are compared.
- truepart: The value that is returned if the comparison is true.
- falsepart: The value that is returned if the comparison is false.

6.12 LoopWhile [lvalue | rvalue | operator | datatype]

Marks the end of a Do...LoopWhile-section and also has the logical test.

- lvalue: The value on the left side of the comparison.
  Legal values:
  1. @currentvalue indicates that the value returned from the previous function.
  2. @othervalue indicates that the value should be retrieved from the variable 'othervalue'.
  3. Values not starting with @ will be used as is.
- rvalue: The value on the right side of the comparison.
  Legal values:
  1. @currentvalue indicates that the value returned from the previous function.
  2. @othervalue indicates that the value should be retrieved from the variable 'othervalue'.
  3. Values not starting with @ will be used as is.
- operator (= | != | >= | <=): The comparison operator.

- datatype (text|integer|money|float|datetime)(optional): The datatype that the values should be when they are compared.

6.13 OrIf [lvalue | rvalue | operator | datatype]

Does a comparison and lets processing continue to the next function if the comparison is true. If the comparison is false, processing will look for endif, else or elseif.

- lvalue: The value on the left side of the comparison.
  Legal values:
  1. @currentvalue indicates that the value returned from the previous function.
  2. @othervalue indicates that the value should be retrieved from the variable 'othervalue'.
  3. Values not starting with @ will be used as is.
- rvalue: The value on the right side of the comparison.
  Legal values:
  1. @currentvalue indicates that the value returned from the previous function.
  2. @othervalue indicates that the value should be retrieved from the variable 'othervalue'.
  3. Values not starting with @ will be used as is.
- operator (= | != | >= | <=): The comparison operator.
- datatype (text|integer|money|float|datetime)(optional): The datatype that the values should be when they are compared.

6.14 While[lvalue | rvalue | operator | datatype]

Loops a set of functions if the comparison is true.

- lvalue: The value on the left side of the comparison.
  Legal values:
  1. @currentvalue indicates that the value returned from the previous function.
  2. @othervalue indicates that the value should be retrieved from the variable 'othervalue'.
  3. Values not starting with @ will be used as is.
- rvalue: The value on the right side of the comparison.
  Legal values:
  1. @currentvalue indicates that the value returned from the previous function.
  2. @othervalue indicates that the value should be retrieved from the variable 'othervalue'.
  3. Values not starting with @ will be used as is.
- operator (= | != | >= | <=): The comparison operator.
- datatype (text|integer|money|float|datetime)(optional): The datatype that the values should be when they are compared.

7. Variables

7.1 SetVariable [name | value | datatype]

Sets a variable value.

- name: The name of the variable.
- value (optional): The value to be set into the variable. If the value is not set, the value from previous functions will be used.
- datatype (text|integer|money|datetime|float)(optional): The datatype that the value will be stored as.

7.2 OperateVariable [name | value | datatype]

Changes the value of a variable.

- name: The name of the variable.
- value: The value that should be set into the variable. If the value is prefixed with an operator, the variable will be changed according to the operator and value; e.g., +1 will increase the value of the variable by 1. The available operators are: =, +, -, *, /.
- datatype (text|integer|money|datetime|float)(optional): The datatype that the value will be used as.

7.3 GetVariable [name]

Gets a value from a variable.

- name: The name of the variable.

7.4 SetDynamicName [variable | format]

Builds a dynamic variable value. Often used for creating dynamic file and directory names.

- variable: The variable name where the result will be stored.
- format: The format of the result variable. Capital letters inside brackets will be replaced with the following: J - Job Number, C - Counter, Y - Year, M - Month, D - Day, H - Hour, N - Minutes, S - Seconds. The length of the resulting number is given by the number of characters; for instance, [JJ] gives a two-digit job number and [JJJJJ] gives a five-digit one. Mixing different characters inside brackets is not allowed.

8. External resources

8.1 CreateGUID []

Creates a GUID.

8.2 GetUserLogin []

Gets user login.

8.3 GetCounter [name]

Increases and retrieves a counter value from an external file ("counter.dat").

- name: The name of the counter.

8.4 GetJobNo []

Returns the job number.

8.5 GetPartnerValue [key]

Gets a partner value.

- key: The key for the partner value.

8.6 SQLConnect [connectionname | connectionstring | userid | password | autocommit]

Creates a new connection to a database. This function has to be run before an SQLExec can be called. Do not put this function in a loop or repeating circle. If the SQLExec function is being used through the entire document mapping, you may put this function at the beginning of the document structure or in the job manager's pre-processing list.

- connectionname: The name of the connection. This name will be referred to from other functions.
- connectionstring: The connectionstring used for the DBS. This usually contains specific parameters for the different DBMS (DSN=MyDB;SERVER=MyDB;UID=DiCentral;PWD=DiCentral;).
- userid: The user ID for the DBS to be connected. Note: Do not use this parameter if the username is present in the connection string.
- password: The password for the connecting user. Note: Do not use this parameter if the password is present in the connection string.
- autocommit: Should the DB connection use autocommit?

8.7 SQLDisconnect [connectionname]

Disconnects an active connection to a database. This function has to be run after the SQLExec is done for a map. Do not put this function in a loop or a repeating circle. If the SQLExec function is being used through the entire document mapping, you may put this function at the end of the document structure or in the job manager's post-processing list.

- connectionname: The name of the connection.

8.8 SQLExec [connectionname | sql | fieldseparator | recordseparator]

Executes an SQL statement on a database. The function should refer to a connection name created by a successful SQLConnect. Since this function is used on a field, make sure the result of the SQL statement returns only a single value and not an entire record set.

- connectionname: The name of the connection.
- sql: The SQL statement you want to run. The returned value will be the new current value for the field. You may put the SQL in a variable, and then set the SQL parameter to the "@variablename". You cannot refer to a variable within the string as a part of the SQL statement.
- fieldseparator (optional): If the SQL statement returns several fields, the field separator will be entered between the fields. If there is no field separator, only the first field will be returned.
- recordseparator (optional): If the SQL statement returns several records, the record separator will be entered between the records. If there is no record separator, only the first record will be returned.

9. Debug

9.1 Debug [variable | tracefile | traceall]

Indicates that the following functions should send full information to the debug-log. If no parameter is set, e.g., no variable name, the content of @currentvalue is reported.

- variable: The name of a variable to monitor.
- tracefile (optional): The file name to put trace information into.
- traceall (y|n)(optional): Y – Trace all nodes. N – Trace current node only. Default is 'N'.

9.2 SaveExd [exd | filename]

Saves an EXD to a file in XML format.

- exd: The name of the EXD.
- filename: The name of the file to save it into.

9.3 LoadExd [exd | filename | category]

Loads an EXD from a file in XML format.

- exd: The document reference of the EXD description (filename.xml#exdname).
- filename: The name of the file to load from.
- category: The category that the new EXD should have.

9.4 SetError [message | severity]

Sets a syntax error. This function will create an error that will stop the processing of the current message.

- message: The message that will appear in the error log.
- severity (debug|info|warning|error|fatal): The severity of the message. Default is ‘error’.

9.5 Validate [operator | rvalue | datatype | severity | message]

Validates current field for business logic.

- operator (= | != | >= | <=): The comparison operator.
- rvalue: The value to check against.
- datatype (undefined|text|integer|money|float|datetime)(optional): The datatype that values should be when they are compared.
- severity (info|warning|error|fatal): The severity of the validation error. Default is ‘warning’.
- message: The message that should be sent when the validation fails.

9.6 ValidateBetween [fromvalue | tovalue | datatype | severity | message]

Validates current field for business logic.

- fromvalue: The lower boundary.
- tovalue: The upper boundary.
- datatype (undefined|text|integer|money|float|datetime)(optional): The datatype that the values should be when they are compared.
- severity (info|warning|error|fatal): The severity of the validation error. Default is ‘warning’.
- message: The message that should be sent when the validation fails.

9.7 ValidateRegex [regex | severity | message]

Validates the current field for business logic, using a regular expression.

- regex: The regular expression to check for a match.
- severity (info|warning|error|fatal): The severity of the validation error. Default is ‘warning’.
- message: The message that should be sent when the validation fails.